

AD-A284 076



1

REPRESENTING VISIBILITY  
FOR SITING PROBLEMS

By

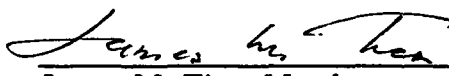
Clark K. Ray

A Thesis Submitted to the Graduate  
Faculty of Rensselaer Polytechnic Institute  
in Partial Fulfillment of the  
Requirements for the Degree of  
DOCTOR OF PHILOSOPHY

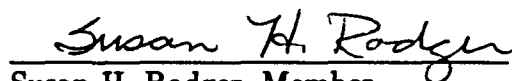
Major Subject: Computer and Systems Engineering

Approved by the  
Examining Committee:


  
Wm. Randolph Franklin, Thesis Adviser

  
James M. Tien, Member

  
George Nagy, Member

  
Susan H. Rodger, Member



16598  
94-28489  


This document has been approved  
for public release and sale; its  
distribution is unlimited.

Rensselaer Polytechnic Institute  
Troy, New York

April 1994  
(For Graduation May 1994)

94 9 01 086

DTIC QUALITY INSPECTED 1

29 August, 1994

Defense Technical Information Center  
ATTN: Selection Section (DTIC-FDAC)  
Building 5, Cameron Station  
Alexandria, VA 22304-6145

Under the provisions of AR 621-1, I recently finished Army fully-funded graduate study leading to a Ph.D., for which a thesis was required. Per Army requirements, I am furnishing two copies of this thesis (enclosed).

Thesis Title: Representing Visibility for Siting Problems  
Name: Clark K. Ray  
Rank: Lieutenant Colonel  
Date: April 1994

Should any additional information be required, I can be contacted at:

LTC Clark K. Ray  
Department of Electrical Engineering and Computer Science  
US Military Academy  
West Point, NY 10996

EMAIL: ray@eecs1.eecs.usma.edu  
Phone: DSN 688-3029  
Commercial 914-938-3029

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Availability Codes
A-1	

*Clark K. Ray*  
CLARK K. RAY  
LTC, EN

94 9 01 086

## CONTENTS

LIST OF TABLES . . . . .	vi
LIST OF FIGURES . . . . .	vii
1. INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	1
1.2 Nature of Computation Costs in the Visibility Problem Domain . . . .	2
1.3 Methodology . . . . .	4
1.3.1 Construct Methods to Access Data . . . . .	4
1.3.2 Elevation Model and Problem Setting . . . . .	5
1.3.3 Viewshed Algorithms . . . . .	5
1.3.4 Visibility Index Algorithms . . . . .	5
1.3.5 Accuracy and Performance Evaluation . . . . .	7
1.3.6 Potential Approach . . . . .	7
1.4 Issues and Goals . . . . .	7
1.4.1 LOS Calculation Constraints . . . . .	7
1.4.2 Performance Improvements over Existing Algorithms . . . . .	8
1.4.3 Empirical Validation of Algorithms on Real Terrain . . . . .	8
1.4.4 Identification of Candidate Point Sites . . . . .	8
1.4.5 Exploit Parallel Processing to Compute Visibility Indexes . . . .	9
1.4.6 Demonstration to User Community . . . . .	9
1.5 Organization of Thesis . . . . .	10
2. LITERATURE REVIEW . . . . .	12
2.1 Background . . . . .	12
2.1.1 Visibility . . . . .	12
2.1.2 Parallelism . . . . .	13
2.1.3 Siting . . . . .	13
2.1.4 Terrain Elevation Data . . . . .	14
2.2 Characteristics of the Problem and its Domain . . . . .	14
2.3 Practical Considerations . . . . .	15

3. VISIBILITY PROBLEM TAXONOMY AND ELEVATION MODEL . . .	17
3.1 Taxonomy of Visibility Analysis Characteristics . . . . .	17
3.1.1 Terrain Characteristics . . . . .	18
3.1.2 Elevation Data Characteristics . . . . .	18
3.1.3 Selection of Sample Observer Points . . . . .	20
3.1.4 Elevations for Non-Data Point Locations . . . . .	22
3.1.5 Uncertainty in the Data . . . . .	22
3.1.6 Observer, Target, and LOS Characteristics . . . . .	23
3.1.7 Calculation Characteristics . . . . .	24
3.1.8 Placement of this Research within Taxonomy . . . . .	26
3.2 Elevation Model . . . . .	26
3.3 Characteristics of an Elevation Dataset . . . . .	32
3.4 Considerations in LOS Calculation . . . . .	33
3.5 Summary . . . . .	36
4. AREA VISIBILITY DETERMINATION—THE R2 ALGORITHM . . . .	37
4.1 Overview . . . . .	37
4.2 Algorithm Considerations . . . . .	39
4.3 R2 Algorithm Development . . . . .	46
4.4 Validity of R2 Algorithm . . . . .	51
4.5 Initial R2 Execution Time Results . . . . .	55
4.6 Summary . . . . .	56
5. DEVELOPMENT OF LOS RAY ESTIMATOR ALGORITHMS . . . . .	60
5.1 VL Algorithm . . . . .	60
5.2 Investigating Visibility Index Estimates . . . . .	61
5.3 Hypotheses and Characteristics . . . . .	63
5.4 Pseudo-Fixed Point Optimizations . . . . .	67
5.5 WeightF Algorithm . . . . .	67
5.6 Weight Algorithm . . . . .	68
5.7 Visibility Images . . . . .	69
5.7.1 Magnitude-Only Visibility Images . . . . .	69
5.7.2 Directional Visibility Images . . . . .	70

5.8	Parallelism Considerations . . . . .	70
5.9	Summary . . . . .	72
6.	ACCURACY OF VISIBILITY ESTIMATES . . . . .	73
6.1	Overview of Methodology . . . . .	73
6.2	Performance of the VL Algorithm . . . . .	75
6.3	Relationships to Elevation . . . . .	79
6.4	Improvements to VL Estimation Algorithm . . . . .	83
6.5	WeightF to R2 Correlation as a Function of Ray Density . . . . .	88
6.6	Analysis of Other Locales . . . . .	90
6.6.1	Dijon Area . . . . .	90
6.6.2	Pripyat' Area . . . . .	91
6.6.3	Summary . . . . .	99
6.7	Confidence in Finding Highest Points . . . . .	99
6.8	Impact of Errors in Elevation Data . . . . .	102
6.9	Summary . . . . .	103
7.	EXECUTION TIMES . . . . .	105
7.1	Overview . . . . .	105
7.2	R3 Execution Time Performance Evaluation . . . . .	106
7.3	R2 Execution Time Performance Evaluation . . . . .	107
7.4	WeightF Execution Time Performance Evaluation . . . . .	115
7.5	Summary . . . . .	118
8.	SUMMARY AND CONCLUSIONS. . . . .	119
8.1	Summary . . . . .	119
8.2	Significant Contributions and Accomplishments . . . . .	121
8.3	Future Work . . . . .	123
	LIST OF SYMBOLS . . . . .	127
	GLOSSARY . . . . .	129
	REFERENCES . . . . .	132
	APPENDICES . . . . .	137
A.	Computation and Storage Costs of Visibility Determinations . . . . .	138

B. Formal Definition of the HAWK Siting Problem . . . . .	142
C. Investigation of Elevation-Visibility Relationships . . . . .	146
D. Demonstration to User Community . . . . .	151

## LIST OF TABLES

Table 1.1	Comparative Costs of Increasing Grid Density . . . . .	3
Table 3.1	Placement within Taxonomy . . . . .	27
Table 4.1	Properties of R2 LOS-Height Error. . . . .	52
Table 4.2	Count of Visible Points by R2 and R3. . . . .	52
Table 4.3	Sample Execution Times of R2 and R3. . . . .	55
Table 7.1	R3 Execution Times as a function of the Visibility Radius . .	106
Table 7.2	R2 Execution Times as a function of the Visibility Radius; Separate Fitted Curves for each Group . . . . .	110
Table C.1	Best Points Visibility—Elevation Correlations. . . . .	147
Table C.2	All Points Visibility—Elevation Correlations. . . . .	149

## LIST OF FIGURES

Figure 1.1	Visibility Cells between Latitudes North 34–40 and Longitudes East 124–130. . . . .	6
Figure 3.1	Grids of Varying Point Densities. . . . .	19
Figure 3.2	Elevation Histogram of DTED Cell N37E127 . . . . .	32
Figure 3.3	Log Scale Elevation Histogram of DTED Cell N37E127 . . . . .	33
Figure 3.4	Example of a LOS Profile. . . . .	34
Figure 4.1	Octant Divisions. . . . .	37
Figure 4.2	Logical View of First Octant . . . . .	38
Figure 4.3	LOS Crossings of Grid Lines. . . . .	40
Figure 4.4	Progression of LOS Wave Front. . . . .	42
Figure 4.5	Erroneous Propagation in Wave Front Algorithms. . . . .	43
Figure 4.6	Propagation of Maximum Chunk Distortion LOS-Height Error with Distance from the Observer. . . . .	45
Figure 4.7	Operation of the Basic R3 Algorithm. . . . .	46
Figure 4.8	Coverage Problems in the Initial R2 Algorithm. . . . .	48
Figure 4.9	Modified R2 Algorithm. . . . .	50
Figure 4.10	Distribution of R2 LOS-Height Differences (meters). . . . .	51
Figure 4.11	Comparing 4 Variants of R2 . . . . .	53
Figure 4.12	Agreement of R2 and R3 Values for 1041 Points in N37E127 . . . . .	54
Figure 4.13	Alternative Tessellation of R2 Observations. . . . .	57
Figure 4.14	De-Cluttered Alternative Tessellation from R3. . . . .	57
Figure 5.1	40-kilometer Visibility Index Values, 32 and 128 LOS Rays, N37E127. . . . .	62
Figure 5.2	Gray-Scale Elevation Image of DTED Cell N37E127. . . . .	64



Figure 5.3	Gray-Scale Visibility Image from DTED Cell N37E127. . . . .	64
Figure 5.4	Log Distribution of LOS Index Values from DTED Cell N37E127. . . . .	65
Figure 5.5	Point-Matched Elevation and Visibility Values from DTED Cell N37E127. . . . .	66
Figure 5.6	Elevation and Visibility, 110 Best VL Visibility Points, N37E127. . . . .	66
Figure 6.1	Spatial Distribution of 1024 Pseudorandom Points. . . . .	75
Figure 6.2	N37E127 - VL and R2 Visibility for 1024 Random Points. . . . .	76
Figure 6.3	N37E127 - Fit Residuals, VL and R2 Visibility, for 1024 Ran- dom Points. . . . .	77
Figure 6.4	N37E127 - VL and R2 Visibility for Top 256 VL Points. . . . .	78
Figure 6.5	N37E127 - Elevation and R2 Visibility for 1024 Random Points. . . . .	79
Figure 6.6	N37E127 - Elevation and R2 Visibility for Top 256 VL Points. . . . .	80
Figure 6.7	N37E127 - Elevation and VL Visibility for Top VL 256 Points. . . . .	81
Figure 6.8	N37E127 - Elevation and VL Visibility for Top VL 100 Points. . . . .	82
Figure 6.9	N37E127 - VL and WeightF Visibility for Top VL 256 Points. . . . .	83
Figure 6.10	N37E127 - WeightF and R2 Visibility for Top VL 256 Points. . . . .	84
Figure 6.11	N37E127 - WeightF and R2 Visibility for 1024 Random Points. . . . .	85
Figure 6.12	N37E127 - WeightF (128 rays) and R2 Visibility for 1024 Ran- dom Points. . . . .	86
Figure 6.13	N37E127 - WeightF (128 rays) and R2 Visibility for Top 256 VL Points. . . . .	87
Figure 6.14	N37E127 - Weight and R2 Visibility for 1024 Random Points. . . . .	88
Figure 6.15	N37E127 - Correlation Variation of R2 to WeightF. . . . .	89
Figure 6.16	N37E127 - Correlation Variation of R2 to WeightF (8 rays). . . . .	90
Figure 6.17	N47E005 - Gray-Scale Elevation Image. . . . .	92
Figure 6.18	N47E005 - WeightF and R2 Visibility for 1024 Random Points. . . . .	93
Figure 6.19	N52E028 - Gray-Scale Elevation Image. . . . .	94

Figure 6.20	N52E028 - Elevation Histogram for 1024 Random Points. . . .	95
Figure 6.21	N52E028 - WeightF and R2 Visibility for 1024 Random Points.	96
Figure 6.22	N52E028 - WeightF and R2 with Alternate LOS Parameters for 1024 Random Points. . . . .	97
Figure 6.23	N52E028 - Histogram of R2 Visibility for 1024 Random Points	98
Figure 6.24	N52E028 - Histogram of R2 Visibility with Alternate LOS Parameters for 1024 Random Points . . . . .	99
Figure 6.25	Center of N37E127 - WeightF Points Required to Find the Highest R2 Points . . . . .	100
Figure 6.26	Center of N37E127 - Histogram of R2 Visibility Values . . . .	101
Figure 6.27	Center of N37E127 - Ratio of WeightF Points Required to Number of Top R2 Points . . . . .	102
Figure 6.28	N37E127 - R2 Visibility for Original and Perturbed Elevations	103
Figure 7.1	N37E127 - 8 Random Points; Weighted Curve Fit . . . . .	107
Figure 7.2	N37E127 - Points within R2 Radius . . . . .	108
Figure 7.3	N37E127 - R2 Execution Times for 256 Random Points . . . .	109
Figure 7.4	N37E127 - R2 Execution Times for 256 Random Points (lower range) . . . . .	112
Figure 7.5	Comparison of R3 and R2 Execution Time for 256 Points. . .	112
Figure 7.6	N37E127 - R2 Execution Times for 256 Random Points (upper range) . . . . .	113
Figure 7.7	N37E127 - WeightF Execution Times for 16 runs of 256 Pseudo- Random Points (out to 60 kilometers) . . . . .	116
Figure 7.8	N37E127 - WeightF Execution Times for 16 runs of 256 Pseudo- Random Points (from 70 to 90 kilometers) . . . . .	117
Figure 8.1	Elevation Vicinity n37e126y . . . . .	124
Figure 8.2	Visibility Vicinity n37e126y . . . . .	124
Figure D.1	Screen from PC Demo Application. . . . .	152

## ACKNOWLEDGMENT

I would like to thank my thesis advisor, Professor Wm. Randolph Franklin, and the members of my committee, Professor James M. Tien, Professor George Nagy, and Professor Susan H. Rodger for their patience and guidance during the course of my research.

Partial support for this work was provided by the Directorate for Computer and Information Science and Engineering, National Science Foundation grant CDA-8805910, and by National Science Foundation grant CCR-9102553.

Part of this work was supported by the U.S. Army Topographic Engineering Center (Mr. John R. Benton) under the auspices of the U.S. Army Research Office Scientific Services Program administered by Battelle (Delivery Order 2756), contract DAAL03-86-D-0001.

I am deeply appreciative for the equipment and other support provided by the Department of Electrical Engineering and Computer Science, U.S. Military Academy.

I am also very grateful for the support provided by the Fannie and John Hertz Foundation during significant portions of my graduate experience, and to the Free Software Foundation and other contributors to the body of freely available software tools.

Finally, I can thank but never repay the members of my family for their understanding and support, and their belief in me regardless of the circumstances.

## ABSTRACT

Many siting and other terrain analysis problems require the determination of the visibility from large numbers of observer locations over their surrounding terrain. The computational costs of existing visibility algorithms often require serious compromises on the fidelity, accuracy, or scope of analysis in real world settings. A highly accurate viewshed determination algorithm, R2, is presented that executes in  $O(R^2)$  time (where  $R$  is proportional to the number of points along the radius of the potential viewshed) and does not suffer from numerical problems, complex special cases, or high constant factors. Since the size of the output in viewshed determination is also  $O(R^2)$ , the time complexity of R2 is within a constant factor of optimality.

The R2 algorithm is used to verify a fast and effective visibility index estimation procedure, WeightF. WeightF has time complexity  $O(R)$  and produces visibility index estimates with a correlated variation to R2 in excess of 0.9 on a variety of terrain datasets and problem settings. These visibility indexes can be used to select points with best aggregate visibility in an area. On real terrain, analysis using WeightF shows that points with significantly above average visibility index values are only a very small number of the total observer points. This property and the fast execution time of WeightF makes it useful for identifying high visibility points for use in siting algorithms, and to produce a viewable visibility index image that can display a large amount of visibility information in an intuitive fashion.

## CHAPTER 1

### INTRODUCTION

#### 1.1 Motivation

The explosion of capabilities for collecting and distributing digital elevation data for terrain from around the world has led to a significant increase in the amount of computation required to exploit this information for problems where visibility is an important factor. Use of naive visibility determination algorithms results in much greater than linear increases in computation requirements as elevation dataset sizes increase.

The increasing power of widely available workstations and personal computers invite exploitation of increasing volumes of terrain data. However, the straightforward use of existing visibility analysis procedures either degrades a significant amount of the information present in the original elevation data or imposes computational requirements far in excess of current or anticipated hardware capabilities when examining even modest real world problems. Various existing approximation methods reduce this complexity but can suppress the effects of relatively minor variations in terrain elevation that may have a major impact on visibility. Careful design and selection of visibility estimating algorithms and representations can make exploiting elevation data more computationally tractable.

Intrinsic to *visibility* is the concept of a *line-of-sight*. The term line-of-sight (LOS) refers to a path connecting an *observer* and the point being observed (the *target*). To determine a LOS, the location of the observer and the target must be known in three dimensions – the two-dimensional geographic location and height above some common referent. A LOS may be an essentially straight line, as in the case of an optical LOS, or (for many electromagnetic signals) may ‘bend’ around

irregularities in the terrain. If the LOS between two points is not obstructed by terrain or other interference, it is referred to as *clear*. If some obstruction prevents observation between the two endpoints of a LOS, it is considered *blocked*.

One important consideration in the problem of siting certain kinds of facilities (such as radar and communication antennas) is the quantity and location of areas visible from candidate sites. Many practical problems require maximizing the total value of covered sub-regions without requiring complete coverage of the geographic area of interest. The details of a specific siting problem are described in Appendix B. In real world cases, such problems are characterized by insufficient facilities (e.g., air defense units) to provide complete coverage of all important facilities within the Area of Interest (AOI) and the need to select and evaluate potential siting combinations in an on-line fashion.

Traditional methods of comparing visibility from different locations become difficult to use when more than a trivially small number of locations is being considered. Non-traditional methods of portraying visibility information for on-line analysis might prove to be of great benefit to the process of interactive analysis and decision making [Ray94].

## 1.2 Nature of Computation Costs in the Visibility Problem Domain

Storage requirements increase with the square of the linear density of data points per unit distance on actual terrain. The terrain model employed here, based on gridded elevation data, is described in Chapter 3. For a specific ground distance radius, the computation cost for determining the visibility to all surrounding points from a given observer point by naive methods increases with the cube of the linear density of data points. The computation cost for determining the visibility to all surrounding points from all potential observer points in a given ground area increases with the fifth power of the linear density of data points.

An example<sup>1</sup> using representative problem parameters illustrates the impact of point density on computational cost. The costs for data storage and computation to calculate visibility from each point in an area approximately 100 kilometers square out to all points up to 40 kilometers distant are as shown in Table 1.1. These results assume elevation values are stored as two-byte integers and that visibility determination to each surrounding point from a given observer point takes on the order of 1 millisecond of CPU time executed on a Sun IPC workstation for the case of 80 meter spacing between points.

<i>Point Spacing</i>	<i>Storage (MB)</i>	<i>Time</i>
1000 meters	0.06MB	67 minutes
300 meters	0.69MB	19 days
80 meters	9.68MB	39 years
30 meters	68.84MB	52 centuries
10 meters	619.52MB	1,274 millennia

**Table 1.1: Comparative Costs of Increasing Grid Density**

Greater elevation point density is desirable to allow more faithful representation of the terrain for which any given analysis is being conducted. However, as Table 1.1 shows, modest increases in the density of data lead to an unacceptable explosion in the cost of computation when attempting a thorough visibility analysis by traditional methods. Significant increases in point density for a variety of raster terrain data have occurred in the past decade, and given the ongoing increases in the ability to collect, store, and access large volumes of data this trend can be expected to continue. For example, digital satellite imagery from the former U.S.S.R. is now available on the commercial market at a resolution estimated at between 1.5 and 2 meters. Low and decreasing prices for GPS (Global Positioning System) receivers and completion of the associated system of satellites will greatly reduce the cost of collecting accurate digital elevation measurements, potentially leading to large volumes of high density data. Sample 1 meter resolution elevation datasets have

---

<sup>1</sup>See Appendix A for details.

already been produced at the U.S. Army Topographic Engineering Center (TEC) using automated digital stereo extraction.

### **1.3 Methodology**

To find suitable methods for performing visibility analyses at reduced computational cost, our general approach is to construct the necessary automated tools to facilitate investigations using real and simulated data; define a formal elevation model and problem setting for algorithm development; develop a computationally tractable and highly accurate algorithm for *viewshed* determination to serve as a benchmark for faster ordinal visibility estimation techniques; develop techniques to rapidly and accurately estimate a relative ordinal value of visibility in large sets of observer points; validate accuracy and predicted performance; and suggest potential applications.

#### **1.3.1 Construct Methods to Access Data**

Digital elevation data is available from several sources, such as the Defense Mapping Agency (DMA) and the U.S. Geological Survey (USGS). Datasets are produced at several resolutions based on different underlying models [USG90], and are typically not organized in a form for efficient access and processing. A necessary preliminary step in researching approaches to extracting and making use of visibility information derived from elevation datasets of significant size is the ability to transparently access several existing formats of wide-coverage elevation data. Tools constructed to provide this access should facilitate construction of higher levels of abstraction as appropriate algorithms and representations are developed.



### 1.3.2 Elevation Model and Problem Setting

In general, lines-of-sight between two locations will pass between points of known elevation, not just directly over them. This requires that a strategy must be selected for approximating the elevations of points not explicitly represented in the data. On a rectilinear grid where the use of higher order than linear interpolation methods is not warranted, we can establish data point selection guidelines for designing and evaluating line-of-sight determining algorithms.

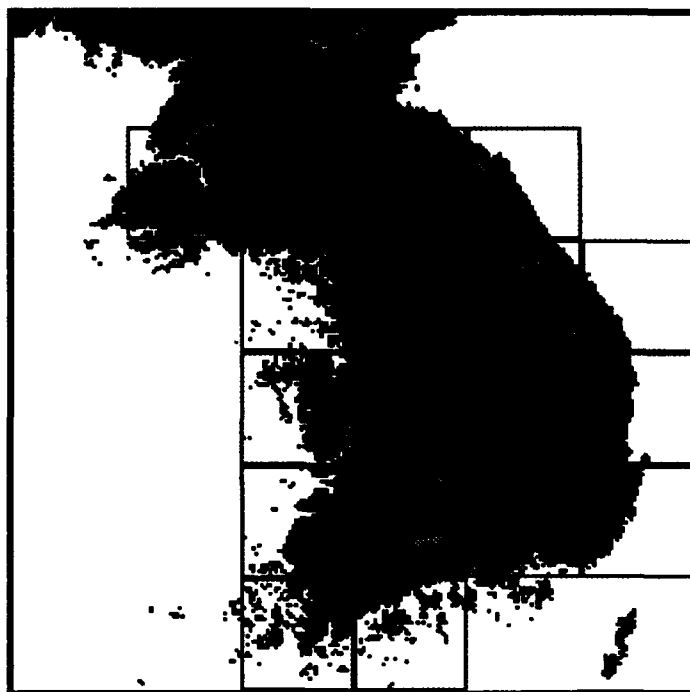
### 1.3.3 Viewshed Algorithms

A fundamental objective of this research is to construct and validate algorithms that provide substantially improved levels of performance (required computation) for given levels of accuracy.

Exhaustive, non-approximate determination (referred to as the R3 algorithm) of the visibility from each grid point within the radius of interest from an observer point varies with the cube of the linear density of elevation data points. An algorithm (referred to as the R2 algorithm) for convex areas of interest was developed whose complexity varies with only the square of the linear density of points while producing high agreement (more than 90 percent) with the results of the R3 algorithm in determining the visibility of target points from an observer point. Details of the R3 and R2 algorithms are presented in Chapter 4.

### 1.3.4 Visibility Index Algorithms

To derive some measure of visibility for each point in an area of realistic size, the results summarized in Table 1.1 clearly demonstrate some form of estimation is required. Assigning a visibility index to each point in the grid of elevation data produces a corresponding *visibility grid*. Relationships that exist between corresponding elevation grids and the visibility grids need to be identified and exploited.



**Figure 1.1: Visibility Cells between Latitudes North 34–40 and Longitudes East 124–130.**

The process of investigating and refining visibility index estimation procedures was conducted over a variety of terrain datasets in order to explore the broadest applicability and to obtain insights from as many detected special cases as possible. Digital Terrain Elevation Data (DTED) data cells (described in Appendix A) from DMA that have been processed to produce visibility information are shown as the outlined area in Figure 1.1

### **1.3.5 Accuracy and Performance Evaluation**

The R2 algorithm, due to its reduced computational requirements, can be used as a means to evaluate the accuracy of visibility index estimation algorithms. Predicted variations in the computation costs with increasing problem size can be compared to timing results obtained from tests conducted on actual hardware.

### **1.3.6 Potential Approach**

To make the calculation of visibility values for candidate facility sites more tractable, a hierarchical approach can be employed. First, an algorithm with greatly reduced computation load can be employed to examine each possible site to rank order them from worst estimated visibility to best estimated visibility by assigning a *visibility index* to each point. Once this has been accomplished, a more computationally demanding procedure can be used to investigate a limited number of points (those with the highest estimated visibility).

## **1.4 Issues and Goals**

### **1.4.1 LOS Calculation Constraints**

In keeping with the elevation model presented in Chapter 3, any LOS procedures employed should take into account all relevant elevation data but prevent all other elevation data points from influencing visibility results. Stated more formally:

- **Criterion for Adequacy** Every elevation point which is 4-neighbor (east, west, north, or south) adjacent to a line-of-sight should contribute to determining visibility and line-of-sight height for that line-of-sight.
- **Criterion for Appropriateness** Any point which is not 4-neighbor adjacent to a line-of-sight should not contribute to determining visibility and line-of-sight height for that line-of-sight.

#### 1.4.2 Performance Improvements over Existing Algorithms

The execution time of the R2 algorithm compares very favorably with the results from [Jun89] and [Sha90]. Since the total number of locations,  $N$ , is proportional to the square of the linear density of points for a convex area of interest, in the terms of [Jun89] and [Sha90] the R2 algorithm achieves a  $O(N)$  running time with a low constant term, along with a lack of geometric special cases and floating point round-off errors.

#### 1.4.3 Empirical Validation of Algorithms on Real Terrain

Since we will attempt to capitalize on the spatial correlation present in real terrain to reduce the computation required to determine useful estimates of visibility index values, the results of such algorithms are checked and evaluated on real terrain of various compositions and locations. Accuracy is evaluated based on demonstrated utility in finding best visible points or in showing residual visibility uncertainty equal to or less than that due to uncertainties in the elevation data.

#### 1.4.4 Identification of Candidate Point Sites

Single point candidate sites can be selected based on their visibility index values. However, naive selection of the  $n$  best sites over an entire area will produce in some cases highly clustered results in potentially unsuitable areas. What is more

likely to produce useful results is to select the  $m$  best sites in  $k$  subareas of the total area, where  $n = k \times m$ . Some mechanism for estimating the appropriate size of subareas, based on the characteristics of the terrain or the specific siting problem type, must be developed.

A straightforward procedure for forcing the selection of the  $n$  best sites to spread over the entire area of interest by means of subarea histogramming has been developed and demonstrated. This could be extended to use overlapping subareas and employ local terrain characteristics to influence subarea size.

#### **1.4.5 Exploit Parallel Processing to Compute Visibility Indexes**

The independence of the computations involved in calculating the visibility index for a given point (e.g., the rays out from the observer) and for each point in the AOI suggests the applicability of parallel processing to reduce the real time requirements for producing a visibility image of a given area. This has been employed with success using a series of workstations of various capacities belonging to the Department of Electrical Engineering and Computer Science at the U.S. Military Academy at West Point. Disk storage requirements remained modest due to the use of networked file systems. Similar parallelism can be employed in use of the R2 algorithm to refine the estimated selection of best points produced by calculation of the visibility index. Parallel computation at several levels of granularity was employed in conducting visibility analyses, resulting in an increase in throughput approximately linear in the number of processors employed.

#### **1.4.6 Demonstration to User Community**

At an interim point in this research, feedback from users with interests in terrain visibility was sought by distributing preliminary results and soliciting evaluation and comment on the most useful aspects and areas for further investigation.

A demonstration system that can execute on many common MS-DOS personal computers has been constructed and furnished to TEC for evaluation and possible distribution to a terrain user community within the U.S. Army (see Appendix D). This system provides side-by-side presentation of elevation and visibility gray-scale images with predetermined locations of high-visibility sites, along with data (reduced detail 16-to-1) for 12 one-degree cells on the Korean Peninsula. An enhanced version of this prototype may serve as the basis for a component of the next release of the TEC AirLand Battle Environment (ALBE) software suite.

TEC is also using algorithms developed as part of this research as part of their prototype 1 meter spacing elevation dataset analysis project for the U.S. Marine Corps. Applications include on-line viewshed display and visibility image generation. Plans are in progress to incorporate this work into TEC's Terrain Elevation Module and in applications on the Department of Defense Simulation Network (SIMNET).

## **1.5 Organization of Thesis**

This chapter presented an introduction to the general research topic and the methodology, goals, and organization of the thesis.

Chapter 2 provides a survey of previous work on the analysis of elevation data, methods for visibility determination, and approaches to making use of visibility results, such as in siting decisions.

Chapter 3 presents a taxonomy of factors that affect the conduct of visibility analyses and an explicit, detailed model of the elevation data upon which our research investigations will be conducted.

The development of basic and improved algorithms for determining the viewshed from a given observer location is described in Chapter 4.

Chapter 5 describes the rationale, development, and refinement of a fast procedure to determine accurate visibility index estimates for large numbers of observer locations, including considerations for parallel execution.

In Chapter 6, the accuracy of the visibility algorithms developed during the course of this research is analyzed in a variety of problem and terrain settings.

Chapter 7 presents the results of verifying the predicted execution time performance of the viewshed determination and visibility index estimation algorithms.

Chapter 8 summarizes the results and contributions of this research and proposes several avenues for continued work. Among the most significant results are the development and validation of an accurate  $O(R^2)$  viewshed determination algorithm, called R2, and a consistent visibility index estimation algorithm called WeightF that executes in  $O(R)$  time.

## CHAPTER 2

### LITERATURE REVIEW

This chapter presents a summary of pertinent results regarding the visibility problem area and its associated aspects. Where they relate to specific points in other chapters, certain references will be cited in that context.

#### 2.1 Background

##### 2.1.1 Visibility

Approaches to computer analysis of visibility have been examined for decades. A method for determining visibility for air defense models, the Minimum Altitude Visibility Diagram, was presented in [SI71]. Several terrain visibility algorithms with worst case execution times ranging from  $O(N^2\alpha(N^2, N))$  to  $O(N^3\alpha(N^3, N))$  were evaluated <sup>2</sup> in [Jun89], with  $N$  proportional to the number of locations being tested for visibility from a given observer. Empirical evidence was presented that the method with the best worst case time, the Triangle Sort Visibility Algorithm, had an average case execution time of  $O(N)$  but required substantial special condition handling. Noted in [Sha90] was that the impact of the extremely high constant term in the expression for the execution time of the Triangle Sort Visibility Algorithm precluded its use on the datasets studied there (the quantity of elevation data involved corresponded to the size of a DTED Level I cell). As a result, a visibility algorithm with an empirically derived execution time of  $O(N^{1.24})$  was employed. Also presented in [Sha90] was a matrix of visibility counts based on visibility from a 'scenic view' path and corresponding to the grid of elevation values.

Visibility calculations have been used in non-traditional ways, such as for use

---

<sup>2</sup>Where  $\alpha(m, n)$  is the inverse of Ackermann's function and for all 'practical' purposes has an upper limit of 4.



in feature inference [DFFP<sup>+</sup>86], [DFNN88]. [DFJN92] present the use of *visibility counts* for use in image interpretation by performing a visibility analysis on a gray-scaled image by treating the pixel values as elevations and using a standard  $\Theta(R^3)$  per observer visibility algorithm. A current overview of visibility computation issues is presented in [Nag94].

### 2.1.2 Parallelism

Calculating visibility lends itself well to parallelized computation. Medium and coarse grained parallelism in investigating the visibility properties of a large number of observer points can be exploited without special programming or hardware (see Section 5.8). A finer grained approach using hypercube hardware for parallelizing visibility determination from a single observer is presented in [TD92], along with a discussion of some of the common pitfalls possible in designing visibility algorithms. Relative visibility values are often a factor in facility siting problems. The use of a transputer array in a heterogeneous computing environment, including personal computers, to execute a hierarchical location selection algorithm is described in [Den93]. A strategy for integrating parallel computation hardware for use in spatial analysis problems into existing networks and GIS environments is described in [Li93]. [Mow93] presents a case study of implementing spatial analysis procedures on Thinking Machines CM-2 and CM-5 parallel computing hardware, observing that synchronous message passing operations common in fine grained parallelism can significantly reduce overall computational throughput.

### 2.1.3 Siting

Methods of siting a limited number of resources to cover spatially distributed demands have also been studied for some time. Optimal site selection under the relatively simple constraint of a fixed maximum distance or time between a site and

its serviced localities was considered in [TR72]. When each potential site can be associated with coverage of a specific sub-region of the total AOI, determining the minimum number of sites required to cover the entire region can be stated as the Minimum Cover Problem [GJ79] which is known to be NP-Complete unless each potential site covers two or fewer 'points' (distinct locations). The Maximal Covering Location Problem as formulated in [CR74] incorporates maximizing coverage of a population and presents several heuristics and a linear programming formulation of solutions. [Mil93] presents a three by three location problem typology of nine distinct combinations of point, line, and polygon natures of facility and client locations. It was shown in [DFFP<sup>+</sup>86] that a standard set-covering algorithm can determine the minimal set of observation points from which an entire surface can be inspected, using a triangulated terrain model and determining the visibility regions of the vertices to provide the required inputs.

#### **2.1.4 Terrain Elevation Data**

DTED Level I cells of elevation data typically represent a geographic area measuring one degree of latitude by one degree of longitude. Such cells are often referred to by the geographic coordinate of their southwest corner. For example, 'N37E127' refers to a cell of approximately 1.5 million elevation points in a 1201 by 1201 matrix. This cell represents terrain between latitudes 37 and 38 north and longitudes 127 and 128 east. DTED Level I datasets (cells) have been produced for large portions of the earth's total land area during the past 15 years.

## **2.2 Characteristics of the Problem and its Domain**

While many approaches have been made to the problem of determining or using visibility characteristics, such as Triangulated Irregular Networks (TIN) [PDDT92], Blocking Edges [Nai88], and Visibility Dominance [Lee92], none seem to translate

well to the scope of current real world problems and datasets. Existing approaches tend to become too computationally costly as the size and density of data increase, or involve polygonal approximations of the available data resulting in loss of information significant enough to obscure the true relative visibility merit of some potential sites. Results presented in [Kum92] contradict the notion of storage space advantages to TINs over DEMs, as do cases cited in [Lay93]. Advances in the capabilities and availability of on-line data compression and decompression technologies may further the relative space efficiencies of DEM representations.

### 2.3 Practical Considerations

The need to clearly describe the models and algorithm implementations used in visibility studies and GIS systems in general has been stated in [Fis93]. In many commercial GIS systems, the algorithms and assumptions employed in viewshed and other calculations are often not revealed due to commercial competitive or liability considerations. This amplifies the potential utility of a set of efficient, experimentally investigated, freely available visibility tools.

The general data model used here will be based upon a raster representation of the elevation attributes of a terrain surface. Other representations exist and are in common use, such as Triangulated Irregular Networks (TIN) and elevation contours. An overview of TIN's and the use of Delaunay triangulation to generate them from grid cell data is given in [Tsa93].

It is unavoidable that in attempting visibility determinations there will be errors introduced in attempting to represent physical terrain with a model and finite digital representation. [Goo92a] distinguishes between *error descriptors* and *error models*. Error descriptors are measures of the difference between the value of a model quantity and the ground-truth value. An error model forms the basis for a stochastic process that can generate a population of sample instances of true

terrain values that differ only in the distortions introduced by error. Generally, error descriptors can be thought of as pertaining to inputs and error models help gauge outputs.

Errors in DEM values can be distributed randomly or systematically. The RMSE for many visibility analyses is often taken as about 7 meters [Fis92], based on a common USGS standard. [Lay93] mentions two common types of systematic DEM errors, spikes and strips, with the nature of the errors in a specific DEM influenced by the methods used in its creation; strip error magnitudes of 10 meters were reported. Aspect (directional, not gradient magnitude) measures were reported as highly sensitive to the influence of random errors in DEMs.

## **CHAPTER 3**

### **VISIBILITY PROBLEM TAXONOMY AND ELEVATION MODEL**

In this chapter we first present a detailed taxonomy of the specific characteristics of the type of visibility analyses we will be addressing. We then present a model of the elevation values (based on raster elevation datasets) that will be used in designing the desired viewshed and visibility algorithms, along with a description of the general nature of LOS determination.

#### **3.1 Taxonomy of Visibility Analysis Characteristics**

Although a great deal of work has been done on various ways of examining the general problem of determining visibility information from digital terrain data, it is not always easy to directly compare the details and conclusions of various investigations. In many analyses of typical GIS operations, the specific assumptions and methods are not consistently, clearly, and completely stated [Kna92]. The degree of general applicability from any given analysis may be significantly impacted by the problem parameters used; for example, an analysis may have been conducted for a relatively short visibility range for only a few points in a heterogeneous terrain setting.

To lay out a framework for developing and analyzing the various visibility algorithms presented in later chapters, as an initial step it is necessary to explicitly list some of the important dimensions of visibility problems. There will be no attempt to claim that the specific results to be presented will span all of the characteristics described below, but by clearly placing the context in which this research resides it will be more straightforward to evaluate it, extend it, and identify contexts in which it is most and least applicable.

### 3.1.1 Terrain Characteristics

- *Relative Complexity* of terrain. The complexity of terrain, for the purposes of visibility determination, is related to the degree and frequency of slope changes. The significance of such changes is dependent on the assumed height of the observer and target above ground level and of the length of the radius of interest.
- *Heterogeneity* in the kinds of surface configuration making up the area being investigated can complicate analysis; for example, the presence of large bodies of water or plains areas adjacent to complex terrain can influence the raw visibility of points on the boundary between regions.
- *Man-Made Features* may have a significant impact on potential visibility due to built-up areas or other construction.
- *Vegetation* can seriously impact visibility and is generally not represented in digital elevation data [Slo93]. Other terrain data can provide vegetation information (when available). Such data may be in a non-raster form, be affected by seasonal variations, and have error characteristics much different from spatially corresponding elevation data.

### 3.1.2 Elevation Data Characteristics

- *Density* of the elevation grid (how many data points are used to represent a given distance along the ground) affects the fidelity of the representation of the physical terrain, the complexity and practicality of the types of analysis which can be conducted, and even the nature of what is being measured. For very dense grids, such as 1 meter between points, simply measuring the elevation at the given location may be adequate. For more sparse grids, such

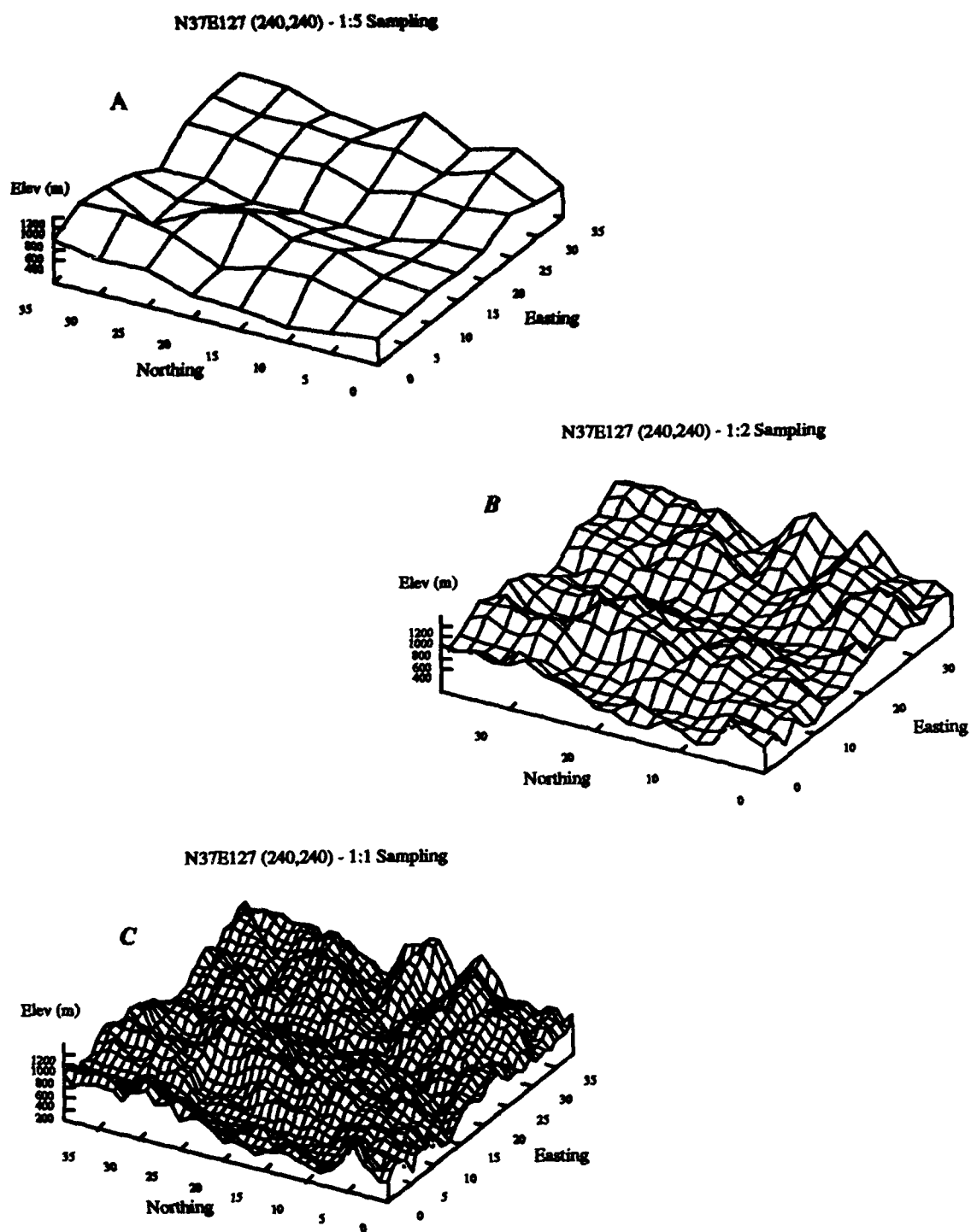


Figure 3.1: Grids of Varying Point Densities.

as 100 meters or greater between points, the data collection agent may have deliberately chosen a 'representative' value to best reflect, according to some (often not explicitly stated) criteria, all of the terrain which is closer to the point being measured than any other grid point. This is an instance of the Modifiable Areal Unit Problem (MAUP) [Fot89]. To provide an intuitive feel for the effect of point density in representing terrain elevation, Figure 3.1 depicts a grid of elevation values taken near cell coordinate (240,240) of DTED cell N37E127 at 1:5, 1:2, and 1:1 samplings.

- *Dataset Size.* Many visibility analyses are performed on datasets of modest sizes, such as one to two hundred points on a side. This limits the amount of computation required and mitigates issues such as earth curvature. However, it limits the ability to conduct large radius visibility analysis, the choice of observer locations (since any observer location with a complete potential viewshed must be no closer than the visibility radius to the edge of available data), and the heterogeneity that can be encountered on real terrain.
- *Earth Curvature* is often ignored in many studies that work with small datasets that cover limited terrain areas [Fis93]. In attempting to work with larger scale problems, however, earth curvature must be accounted for. A close approximation for the variation in effective elevation drop with distance from the observer is

$$\Delta E_C \approx R_E - \sqrt{R_E^2 - D_O^2} \quad (3.1)$$

where  $D_O$  is the distance from the observer,  $R_E$  is the effective radius of the earth, and  $\Delta E_C$  is the change in elevation due to earth curvature.

### 3.1.3 Selection of Sample Observer Points

- *Exhaustive* selection simply chooses every point within an area of interest as



an observer location for visibility analysis. The obvious advantage is complete coverage of the available data (except for the points near the boundaries of the available data for which elevation values are not available out to the radius of visibility in all directions), the relative freedom from the risk of biasing results because of sampling problems, and the large quantity of result values from which to draw inferences. The obvious disadvantages are the potentially large costs in computation time and, depending on the nature of the information collected from the analysis, storage requirements.

- *Arbitrary* selection allows an analyst to investigate hypotheses and perform intuitive exploration, but provides no systematic basis for identifying points that best meet some particular criterion, such as having the highest visibility.
- *Regular Sampling* of the total grid of data points by choosing a subset of points, where the points selected are equally 'spaced' according to some criterion, is an easy way of selecting a sample from a raster of elevation values for the purpose of reducing the amount of data to process when investigating the visibility characteristics of an area. This can produce good coverage spread over the entire area but is less desirable than random samples for the purposes of statistical analysis and may interact with systematic errors in the elevation dataset in unanticipated ways.
- *Pseudorandom* selection of observer locations can meet the dual requirements of spatially distributing a sample of observer locations and preserving the statistical independence of the sample locations from underlying patterns in the elevation data, as long as the pseudorandom number generators are robust.
- *Extreme Value* selection can be used to attempt to match the subset of observer points to particular requirements or investigations. Extreme (high or low) values can be selected or computed based on raw elevation values, visibility

estimates, local gradients, or any measure significantly less costly to calculate than the visibility analysis to be conducted on the sample points themselves.

- *Dispersed Extreme Value* selection employs extreme value selection constrained in some way to ensure spatial dispersion. This can be accomplished by taking the extreme valued point(s) from uniformly distributed subcells, pseudorandom sampling that rejects points that are too close to all existing samples, or other means.

#### 3.1.4 Elevations for Non-Data Point Locations

- *Locations of Defined Elevations.* Possible choices of the set of locations where an elevation value is provided by the elevation model (based on the available data) can include all areas, points on grid intersections and grid line segments (see Section 3.2), data value points only, or a mapping of the available data points to a less dense grid. The TerraBase system [Com88], for example, for some operations maps screen coordinates to geographic coordinates that generally do not fall on grid lines. It uses the four adjacent grid point elevation values to interpolate a value for the desired coordinate.
- *Interpolation Method.* Interpolations can include step functions such as min, max, and nearest; averaging the values of the nearest data points; simple linear interpolation; and higher-order interpolations based on various combinations of points that may use values beyond nearest neighbors.

#### 3.1.5 Uncertainty in the Data

The importance of being aware and explicit about the inevitable *data uncertainty* in the digital elevation information used as the basis for visibility determination has been cited in [WF93] and many other publications. One breakout of data uncertainty issues is:

- *Quantization error* is inherent due to the nature of digital data representation and the essentially continuous variation of elevation on real terrain.
- *Discrete Measurement error* is the difference between the discrete data value closest to the true ground elevation and the value actually present in the digital data; see Equation 3.2
- *Magnitude of errors* assumed present in the data, such as an estimated standard deviation. For analysis in which no error impacts are dealt with, this is implicitly taken as zero.
- *Distribution of errors*, including both a probability density form such as the Gaussian and the degree of spatial autocorrelation (generally taken as zero [Fis92]) presumed present in the errors.

### 3.1.6 Observer, Target, and LOS Characteristics

- *Nature of Observer and Target Locations.* Although commonly taken to be points, the observer and the target can each be represented as a point, a line, or an area. As an example, [TDD91] presents a region (area) to region parallel visibility algorithm.
- *Height of the Observer* above the terrain. In many analyses the height of the observer is taken as zero. This may not be the best choice because it can make the analysis unduly sensitive to minor random, systematic, or discretization errors in the elevation of the observer point and its eight neighbors. In many potential applications of visibility analysis, the observer may be a person standing upright, a facility, or an antenna; none of these take their visibility from ground level.

- *Height of the Target* above the terrain. The height of the target is often taken as zero, although in many cases the purpose of the visibility analysis is to determine where *objects* such as antennas, buildings, aircraft, or eyesores can be seen from the observer. Tops of such objects are generally not taken to be at ground level.
- *Range* from observer to target is often modest both in terms of number of elevation grid points (100-200 on a side) and in the actual ground distances (1-2 km) corresponding to the analysis.
- *LOS Curvature*. The LOS is often assumed to be 'straight', in the sense of an optical LOS with negligible distortions. In problems involving antennas, the ability of an electromagnetic signal to 'bend' around terrain features and detect targets that are not visible to an optical LOS must be considered. Even in these problems, determination of an optical viewshed may be a first step.
- *Tangent Conditions*. Due to the integer representations of elevation common in raster elevation datasets, situations where a computed LOS is exactly tangent to an elevation can occur relatively often. An explicit determination of how to treat such a LOS (clear or blocked at the point of tangency) is necessary.

### 3.1.7 Calculation Characteristics

- *Integer*. Use of integer arithmetic throughout visibility determination is attractive because of its simplicity, speed, relative stability, and match to the problem inputs of integer elevation values. Care must be taken on what intermediate values are represented with integers: for example, if the LOS height above ground at a given point is measured as an integer and carried forward as the basis for the visibility determination for a point further from the observer, accumulated error increases very rapidly.

- *Floating Point* calculations and representations of non-integer values such as slope provide a convenient means of implementing visibility algorithms, but can run substantially slower than integer or fixed point operations on many hardware architectures and introduce complications when testing for equality between values due to round-off error.
- *Fixed Point* arithmetic can combine the speed and discrete comparison nature of integer operations with some of the range and flexibility of floating point operations when the range of values, intermediate results, and accuracy requirements are well understood. Some programming languages (such as Ada) provide built-in support for fixed point operations.
- *Continuous* interpolations provide modeled elevation values for locations which do not have an exact corresponding data point in the elevation raster dataset by means of some continuous function, such as linear or quadratic interpolation.
- *Discrete* approximations provide elevation values for non-data-value points using functions such as min, max, or nearest; the effect of such approximations is typically to propagate an existing elevation data value to some set of points in its vicinity. For example, [TD92] use the elevation data value of a point for visibility computations on any LOS passing through the rectangular cell centered on that point.
- *Number of Calculation Instances.* When we explicitly model uncertainty in the data, we have the option of performing multiple analyses over several instances, or possible realizations, of the elevation data.
- *Representation and Operation Ordering.* The selection of the quantities to be represented in visibility calculations and the order in which arithmetic

operations are applied can have an impact on the outcome of specific visibility results. Just the simple decision to carry forward the integer distance and height values of a visibility-determining elevation instead of a fractional slope value can impact visibility cases where ground elevation is close to the LOS height based on the previous visible point.

### 3.1.8 Placement of this Research within Taxonomy

The models and analyses employed here, unless otherwise noted, are positioned within the visibility taxonomy described in Section 3.1.1 according to the entries in Table 3.1.

Although errors unrelated to quantization in the input elevation datasets are generally not modeled here, see section 6.8 for some preliminary results on the impact on visibility produced by assuming the presence of errors in the input data.

## 3.2 Elevation Model

Let a landscape  $L$  be the physical terrain surface of the AOI defined by a real-valued function  $z_L = f_L(x_L, y_L)$  where  $x_L$  is the longitude,  $y_L$  is the latitude, and  $z_L$  is the height<sup>3</sup> of the surface above mean sea level (elevation) at coordinate  $(x_L, y_L)$ . A tessellation  $T$  of integer values  $e_T = f_T(x_T, y_T)$  at discrete coordinates  $(x_T, y_T)$  approximates the elevation at the corresponding location  $(x_L, y_L)$  in  $L$ . Define discrete measurement error to be

$$d_T = \begin{cases} \lceil e_T - z_L - 0.5 \rceil & \text{if } z_L > e_T \\ 0 & \text{if } z_L = e_T \\ \lfloor e_T - z_L + 0.5 \rfloor & \text{if } z_L < e_T \end{cases} \quad (3.2)$$

This definition of  $d_T$  specifies that there is a non-zero discrete measurement error (in terms of whatever units are being employed, e.g., meters) in the approximation

---

<sup>3</sup>Although the unit of elevation measurement is arbitrary, meters are used in DTED data.

<b>Terrain Characteristics</b>	
<i>Relative Complexity</i>	varied (complex)
<i>Heterogeneity</i>	varied (high)
<i>Man-Made Features</i>	none
<i>Vegetation</i>	none
<b>Elevation Data Characteristics</b>	
<i>Density</i>	70–120 meters (DTED Level I)
<i>Dataset Size</i>	viewshed $\approx$ 750,000 points
<i>Earth Curvature</i>	Equation 3.1
<b>Selection of Sample Observer Points</b>	
<i>Exhaustive</i>	yes
<i>Arbitrary</i>	no
<i>Regular Sampling</i>	no (tools developed)
<i>Pseudorandom</i>	yes
<i>Extreme Value</i>	yes
<i>Dispersed Extreme Value</i>	yes
<b>Elevations for Non-Data Point Locations</b>	
<i>Locations of Defined Elevations</i>	grid points & line segments
<i>Interpolation Method</i>	linear
<b>Uncertainty in the Data</b>	
<i>Quantization</i>	integer (meters)
<i>Discrete Measurement</i>	not modeled
<i>Magnitude</i>	not modeled
<i>Distribution</i>	not modeled
<b>Observer, Target, and LOS Characteristics</b>	
<i>Nature of Observer and Target</i>	points
<i>Height of Observer</i>	5 meters
<i>Height of Target</i>	25 meters
<i>Range</i>	40 kilometers
<i>LOS Curvature</i>	straight (optical)
<i>Tangent Conditions</i>	treat as visible
<b>Calculation Characteristics</b>	
<i>Integer</i>	yes (for LOS calculation)
<i>Floating Point</i>	(only for full weighting)
<i>Fixed Point</i>	yes (for LOS-height calculation)
<i>Continuous interpolation</i>	yes (linear)
<i>Discrete Approximations</i>	no
<i>Number of Calculation Instances</i>	one
<i>Representation &amp; Operation Ordering</i>	32-bit integer intermediate; manually optimized

Table 3.1: Placement within Taxonomy

of  $z_L$  by  $e_T$  if the absolute difference between  $z_L$  and  $e_T$  exceeds 0.5.

Note that for any given  $(x, y) \in T$ , in general  $e_T \neq z_L$  because  $e_T$  is integer valued while  $z_L$  is real valued, or because  $e_T$  may have been chosen by the data collection agency to 'represent' the terrain in the vicinity of  $(x_L, y_L)$  rather than most closely approximate the precise value of  $f_L(x_L, y_L)$ . The discussion of *covering triangles* in this section provides an example of representative elevation selection.

The coordinates  $(x_T, y_T) \in T$  that correspond to coordinates  $(x_L, y_L) \in L$  are typically discrete multiples of some fundamental geographic measurement, such as degrees, minutes, or seconds of longitudinal ( $x_L$ ) or latitudinal ( $y_L$ ) arc. For example, the typical DTED Level I data cell has longitudinal and latitudinal spacing of 3 arc-seconds between 4-neighbor adjacent coordinates. While this method of selecting discrete locations results in a constant latitudinal spacing (approximately 92.6 meters for a 3 arc-second interval), the variation in the longitudinal spacing over such a cell may vary from about 0.02% near the equator to about 2% near north or south 50 degrees latitude. As in many other visibility investigations making use of real world elevation raster data, here the impact of the variation in longitudinal spacing at different latitudes is assumed to be negligible in comparison with other errors resulting from the measurement or sampling process. Variations due to the possible use of ellipsoidal (as opposed to perfectly spherical) surface spheroids in the compilation of geographic data are similarly ignored.

In contrast, elevation variations due to earth curvature cannot be ignored over the distances encountered in many siting problems. For example, over a distance of 40 km the drop due to earth curvature is approximately 125 meters (see Equation 3.1), a distance that is significant in the context of trying to track aerial targets as low as 25 meters above the terrain. Accounting for earth curvature here will usually be accomplished by adjusting the elevation values for each point in the input geographic datasets (which are generally referenced to mean sea level) by the amount



of elevation drop corresponding to the distance between that point and some fixed reference location, such as the center or the southwest corner of the dataset. If other information, such as vegetation height, is available and is significant in determining visibility, it can be incorporated in this step.

Consider a rectangular matrix grid of points  $M$  to be derived from  $T$  in accordance with the assumptions stated above. Every point in  $T$  has a corresponding point in  $M$ . Therefore, if  $T$  is based on a latitude and longitude reference system, so is  $M$ . Since  $M$  is rectangular then for any  $(x_M, y_M)$  in  $M$  without a corresponding  $(x_T, y_T)$  in  $T$  an arbitrary value well outside the range of known actual elevations is chosen as the value at  $M(x_M, y_M)$ . Let  $M_x$  be the number of distinct longitudinal coordinates and  $M_y$  be the number of distinct latitudinal coordinates represented in  $M$ . The total number of points in  $M$  is  $n_M = M_x \times M_y$ . The terrain distance corresponding to all longitudinally adjacent points in  $M$  is a constant,  $m_{sx}$ , and the terrain distance corresponding to all latitudinally adjacent points in  $M$  is a constant,  $m_{sy}$ , for all points in  $M$ . Stated more formally, let  $GD[p_1, p_2]$  be the ground distance (for example, in meters) between points  $P_1, P_2 \in L$ , that corresponds to the two points  $p_1, p_2 \in M$ . Then

$$m_{sx} = GD[(x_i, y_j), (x_{i+1}, y_j)]$$

$$\text{for } i \in \{1, 2, \dots, M_x - 1\}, j \in \{1, 2, \dots, M_y\} \quad (3.3)$$

$$m_{sy} = GD[(x_i, y_j), (x_i, y_{j+1})]$$

$$\text{for } i \in \{1, 2, \dots, M_x\}, j \in \{1, 2, \dots, M_y - 1\} \quad (3.4)$$

Some terrain models (e.g., [TD92]) consider each grid point to be at the center of a rectangular cell and use its value for the elevation of any location within the rectangle whose elevation is required. Other models make provisions for estimating elevation values of locations not coincident with a grid point as some function of two or more points in their neighborhood. In the model presented here the only

locations for which elevation values are known or will be estimated occur either at *grid intersection points*  $M(x_i, y_j)$ ,  $i \in 1 \dots M_x$ ,  $j \in 1 \dots M_y$  or on *grid lines*. Grid lines inside  $M$  are defined by the equations

$$x = i \text{ for } i \in \{1, 2, \dots, M_x\} \quad (3.5)$$

$$y = j \text{ for } j \in \{1, 2, \dots, M_y\} \quad (3.6)$$

A *grid line segment* is a portion of a grid line whose endpoints are grid points in  $M$  that are 4-neighbor adjacent. Generally, the elevation value for a location corresponding to a point on a grid line segment but not on a grid intersection will be estimated by linear interpolation between the values at the endpoints. More sophisticated estimation schemes (splines, etc.) for points on and off the grid lines can be used where specific knowledge of the terrain or application warrant, but this model in its basic form attempts to minimize the assumptions and computation complexity in dealing with commonly available elevation raster datasets.

For dealing with the elevations of points not on grid line segments, we introduce the concept of a *covering triangle*. Consider grid intersection points  $A$ ,  $B$ , and  $C$  in  $M$ , where  $B$  is longitudinally adjacent to  $A$  and  $C$  is latitudinally adjacent to  $A$ . The triangle  $ABC$  is a covering triangle; in this model we assume that the elevation of any point in  $L$  is no higher than the elevation of a corresponding point in a covering triangle. There can be at most two distinct covering triangle elevations corresponding to a particular  $(x_T, y_T)$ , and we assume that  $f_L(x_L, y_L)$  is no higher than a projected point in any corresponding covering triangle. Given this assumption, we can make LOS visibility determinations by examining the intersections of LOS profiles with grid line segments and grid intersection points, and not have to explicitly estimate the elevation of other (non-grid) points in  $L$ .

Using the previous definitions, consider the two-and-a-half-dimension surface  $S$  constructed from the grid points and grid lines in  $M$ . Every location  $(x_L, y_L)$  in  $L$  can be mapped to a corresponding location in  $S$  by adding an appropriate constant

offset  $(OLS_x, OLS_y)$  and scaling by  $m_{sx}$  and  $m_{sy}$ .

$$x_S = (x_L + OLS_x)/m_{sx} \quad (3.7)$$

$$y_S = (y_L + OLS_y)/m_{sy} \quad (3.8)$$

In general,  $x_S$  and  $y_S$  are real valued such that  $0 \leq x_{LS} \leq M_x$  and  $0 \leq y_{LS} \leq M_y$ . Points in  $S$  which are not on grid intersection points or grid line segments are, by assumption, not above their covering triangles and can have no effect on visibility not already modeled by grid segment and grid intersection points. Conceptually, questions about visibility between two points in  $L$  that correspond to points on grid lines or grid line intersections in  $M$  can be analyzed by identifying the corresponding points in  $S$ , performing the analysis using the model definition and available elevation data, and applying the results back to the original points in  $L$ .

The elevation value,  $z_S = f_S(x_S, y_S)$ , where  $0 \leq x_S \leq M_x$  and  $0 \leq y_S \leq M_y$ , is defined as

$$z_S = \begin{cases} M(x_S, y_S) & \text{if } x_S = \lfloor x_S \rfloor, y_S = \lfloor y_S \rfloor \\ \text{INTERPX}(x_S, y_S) & \text{if } x_S \neq \lfloor x_S \rfloor, y_S = \lfloor y_S \rfloor \\ \text{INTERPY}(x_S, y_S) & \text{if } x_S = \lfloor x_S \rfloor, y_S \neq \lfloor y_S \rfloor \\ \text{undefined} & \text{otherwise} \end{cases}$$

where

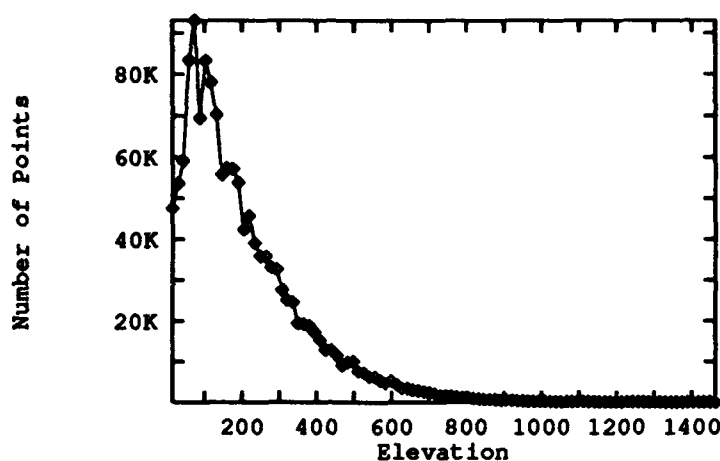
$$\text{INTERPX}(x, y) \equiv M(\lceil x \rceil, y) \times (x - \lfloor x \rfloor) + M(\lfloor x \rfloor, y) \times (\lceil x \rceil - x)$$

$$\text{INTERPY}(x, y) \equiv M(x, \lceil y \rceil) \times (y - \lfloor y \rfloor) + M(x, \lfloor y \rfloor) \times (\lceil y \rceil - y)$$

where 'undefined' means irrelevant for LOS determination purposes (see the discussion of covering triangles, above). Let  $S_{xy}$  be the set of points  $(x, y)$  for which  $z_S$  is defined.

Other estimates such as the maximum, minimum, or average instead of linear interpolation can be used to explore the tradeoffs between and sensitivity of various algorithms.

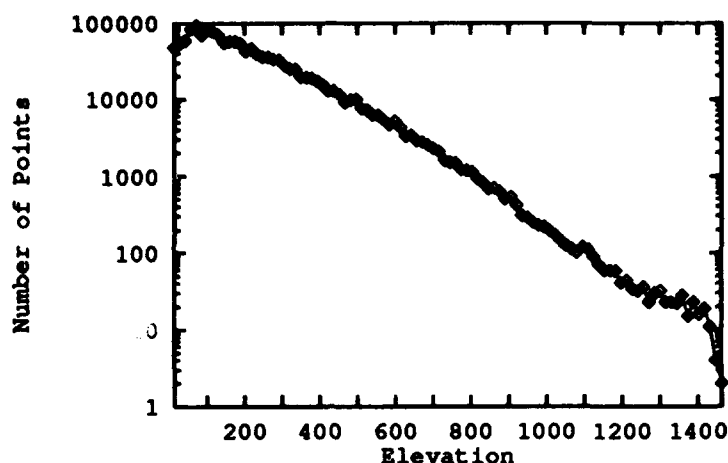
### 3.3 Characteristics of an Elevation Dataset



**Figure 3.2: Elevation Histogram of DTED Cell N37E127**

As a specific example of an elevation dataset, we now consider the empirical properties of the approximately 1.4 million elevation values in the DTED Level I cell N37E127. Figure 3.2 provides a histogram of the elevations for this area, with values ranging from 0 to 1400 meters above mean sea level. In this particular region, clearly most locations are at the lower elevations. An interesting characteristic is revealed when this same histogram is plotted on a log scale as in Figure 3.3.

With the exception of the very extreme values of elevation, the log plot of frequency of points shows a near linearly decreasing profile, suggesting (again, for this particular cell) an approximately exponential decline in the number of points at a given elevation as elevation increases. A similar decreasing exponential relationship for the number of points for given visibility index values is shown in Figure 5.4, although as illustrated in Figures 5.5 and 5.6, and in Chapter 6, there is no apparent strong relationship between high visibility and high elevation points.



**Figure 3.3: Log Scale Elevation Histogram of DTED Cell N37E127**

### 3.4 Considerations in LOS Calculation

The details of calculating a LOS often receive relatively small mention in descriptions of visibility analyses but can have a significant impact on both results and execution times. Even given an elevation data structure as seemingly straightforward as a DTED grid, many interpretations of where and how to calculate LOS information have been employed. In [TD92] calculations used the elevation and location of the nearest grid data point. Some military terrain analysis systems ([Com88]) have evaluated LOS information at points on the interior of an elevation grid cell, requiring an interpolation of elevation from the four surrounding data points. In accordance with the model of elevation presented in this chapter, the LOS methods employed here will only make use of elevation values or estimates on data grid points or grid segments.

Figure 3.4 provides an example of a LOS profile. Shown are observer  $O$  located a distance  $O_h$  above the ground and three potential target locations,  $A$ ,  $B$ , and  $C$ . There are several significant locations, signified by points  $p_0$  through  $p_3$  where certain

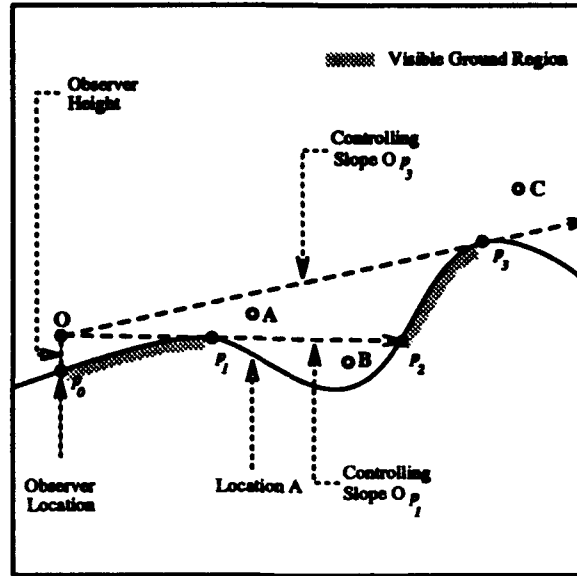


Figure 3.4: Example of a LOS Profile.

LOS's intersect with or are tangent to the ground profile. Intuitively, observer  $O$  will have a clear LOS to all points on or above the ground between  $p_0$  and  $p_1$ , on or above line segment  $\overline{p_1 p_2}$ , on or above the ground between  $p_2$  and  $p_3$ , and on or above the ray from  $O$  through  $p_3$ .

Key in determining visibility over terrain is the *slope* from the observer to points on the ground and to potential target points. Given that the elevation data has been pre-adjusted to compensate for earth curvature, a slope value  $S$  is determined by the familiar

$$S = \frac{\Delta h}{\Delta d} \quad (3.9)$$

where  $\Delta h$  is the difference in the height and  $\Delta d$  the horizontal distance between the endpoints of the line segment defining the slope. Using terminology from the elevation model presented in this chapter, the slope  $S_{\overline{O p_1}}$  from the observer  $O$  to point  $p_1$  in Figure 3.4 is given by

$$S_{\overline{O p_1}} = \frac{f_L(x_{p_1}, y_{p_1}) - (f_L(x_O, y_O) + O_h)}{GD[(x_{p_1}, y_{p_1}), (x_O, y_O)]} \quad (3.10)$$

where  $O_h$  is the height of the observer above ground level and  $(x_O, y_O)$  is the coordinate of the observer projected into the  $xy$ -plane. In Figure 3.4 point  $p_0$  is at ground level directly beneath the observer, i.e.,  $(x_O, y_O) = (x_{p_0}, y_{p_0})$ . Similarly, the slope  $S_{OA}$  from the observer to a target point  $A$  at height  $A_h$  above ground level would be

$$S_{OA} = \frac{(f_L(x_A, y_A) + A_h) - (f_L(x_O, y_O) + O_h)}{GD[(x_A, y_A), (x_O, y_O)]} \quad (3.11)$$

Consider the set of all ground surface points from the observer out to a distance  $R$  along a fixed azimuth. Moving out from the observer, a given point on the ground will or will not be visible from the observer. It will not be visible if the slope to some point closer to the observer is greater than the slope to the given point. This implies that a point can only be visible if the slope to it from the observer is equal to or (more typically) greater than the slope from the observer to *every* ground point between it and the observer. It follows that any ground point further out from the given point can only be visible if the slope from the observer to that point is greater than or equal to the slope to the given point. Thus, any visible ground surface point  $P$  determines a *controlling slope*, equal to the slope of a line segment from the observer to  $P$ , such that if a line with that slope is projected out from the observer to a distance  $R$  then no point beyond  $P$  which falls below that line can be visible from  $O$ .

In the context of Figure 3.4 the initial controlling slope is  $-\infty$ , which is the slope from the observer  $O$  to point  $p_0$ . Moving further out (to the right in Figure 3.4), each subsequent ground surface point defines a new controlling slope until  $p_1$  is reached. Point  $p_1$  is visible from  $O$  so by definition the controlling slope  $S_{p_1}$  at  $p_1$  is the slope of line segment  $\overline{Op_1}$ . The ray  $\overrightarrow{p_1p_2}$  has slope  $S_{p_1}$ , so all surface points between  $p_1$  and  $p_2$  are *not* visible from  $O$ , and  $S_{p_1}$  is the controlling slope between  $p_1$  and  $p_2$ . The surface points  $(x_A, y_A)$  and  $(x_B, y_B)$  beneath potential target points  $A$  and  $B$  are between  $p_1$  and  $p_2$ . Since  $S_{OA} > S_{Op_1}$ , point  $A$  is visible from  $O$ .

Similarly, because  $S_{OB} < S_{Op_1}$ , point  $B$  is not visible from  $O$ .

Since the entire surface line segment between  $p_2$  and  $p_3$  is visible, each successive point from  $p_2$  to  $p_3$  defines a new (and progressively increasing) controlling slope. No points beyond  $p_3$  in this example are visible, so  $S_{Op_3}$  remains the controlling slope to the right of  $p_3$ . Point  $C$  is beyond  $p_3$  and  $S_{OC} > S_{Op_3}$ , so point  $C$  is visible from  $O$ .

Although as abstractions the components of LOS and visibility determination appear straightforward, once we begin to address specific data models and realizable execution environments then correctness and efficiency considerations begin to make themselves felt. To some degree, common data models simplify the theoretical problem by restricting elevation values to integers of a known range. For most models, however, the distance from the observer to various points along a LOS cannot always be represented by integer values. Referring to Figure 4.3, we can observe that although the ground distance between successive x-crossings or successive y-crossings along a given LOS is constant and therefore scalable to integer values, the ratio of all y-crossing to x-crossing interval distances is not. Slope values that are quotients of elevation differences and ground distances (regardless of whether in integer or real domains) are usually real valued, although when elevations and ground or grid distances can be represented as integers then slope values can, of course, be represented as the quotient of two integers.

### 3.5 Summary

This chapter has presented the problem setting taxonomy and elevation model that will be used to develop and validate the desired viewshed and visibility index algorithms, as well as to provide a specific basis to interpret the results of those algorithms. The general nature of LOS determination based on elevation profile data was also described.



## CHAPTER 4

### AREA VISIBILITY DETERMINATION—THE R2 ALGORITHM

#### 4.1 Overview

The R3 algorithm provides a straightforward but relatively costly  $\Theta(n_R^3)$  method of determining for which points within a fixed radius  $R$  or an average radius  $R_{ave}$  of some observer point  $p_o$  a target at height  $h_t$  would be visible to an observer at height  $h_o$  above  $p_o$ . It operates by determining a LOS to each potentially visible point in  $S$  corresponding to a grid intersection point in  $M$ . Elevations are estimated as necessary to fulfill adequacy and appropriateness criteria by whatever means are assumed by the data model, such as linear interpolation (see Chapter 3). In this chapter we will develop and validate a suitable viewshed determination algorithm, R2, that executes in  $\Theta(n_R^2)$  time.

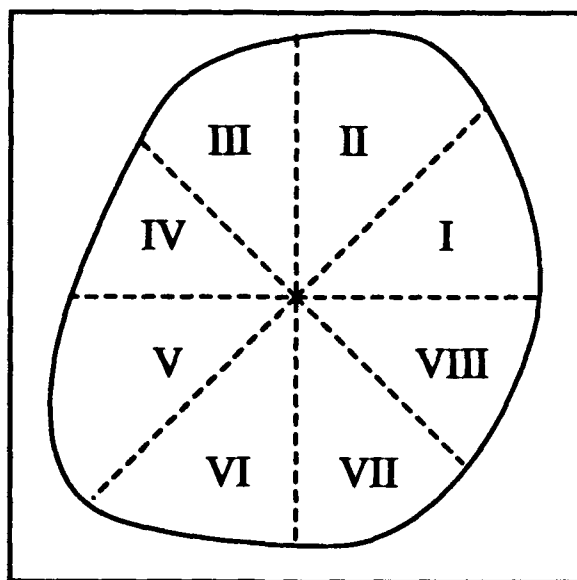
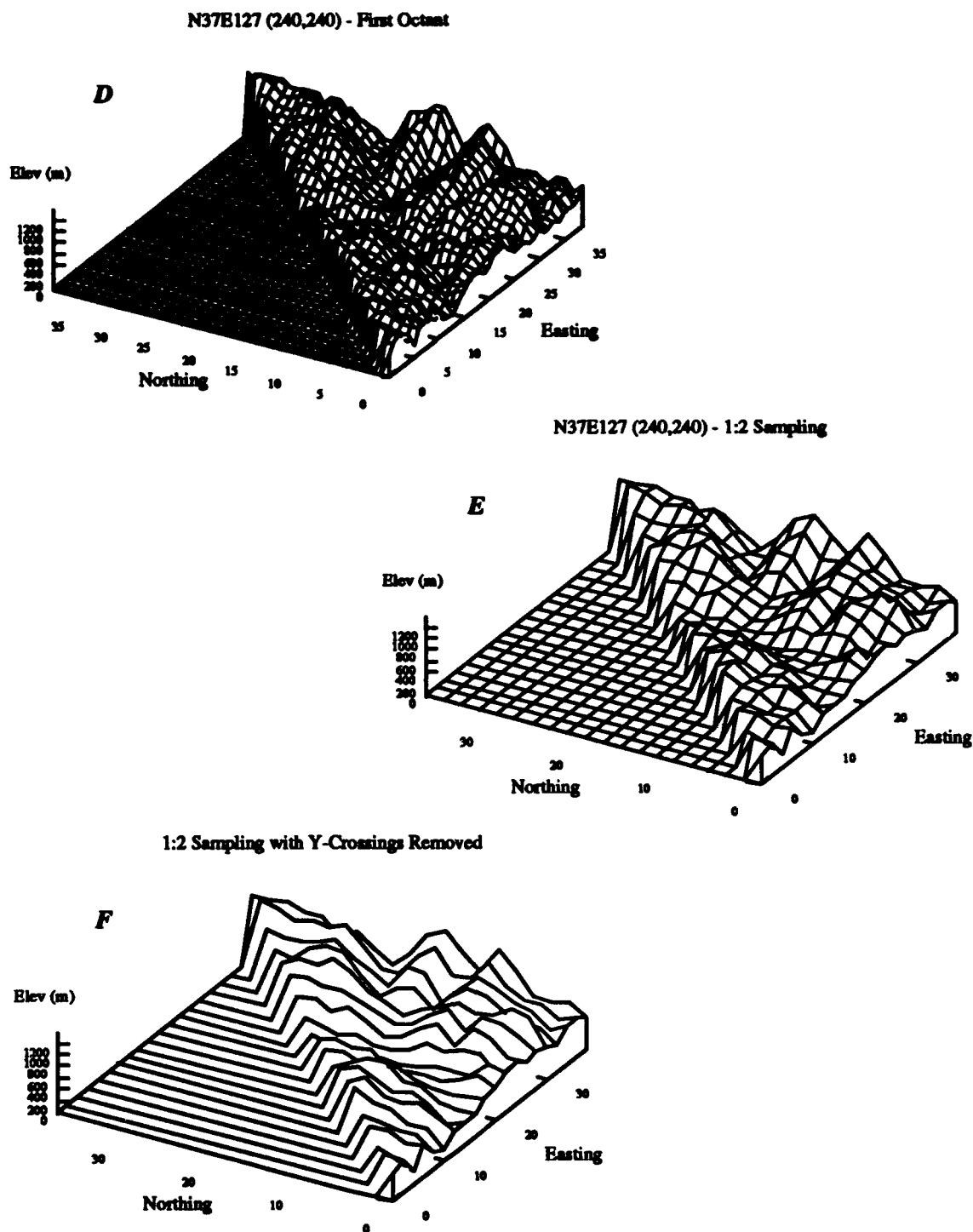


Figure 4.1: Octant Divisions.



**Figure 4.2: Logical View of First Octant**

To provide a frame of reference for discussing certain aspects of constructing practical algorithms for LOS determination, Figure 4.1 shows the numbering system to be used in referring to specific octants. The definition of an octant does not include points that are on the boundaries between octants, which fall on lines making angles of  $k\pi/4$  with the x-axis, where  $k$  is an integer. Chapter 3 presents some of the efficiency and correctness issues that can arise and must be addressed by algorithms employed to compute a single LOS between two points using the data model employed here. Figure 4.2 shows the views of just the first octant elevations for the same elevation data samplings shown in Figure 3.1 earlier.

In the following discussion it is again assumed that the observer point is located within (not on the boundary of) a convex region of target points, which is often circular or elliptical. We consider the case of determining visibility from a single observer point, as opposed to a possibly related group of observer points. Under these conditions, in the absence of further information or assumptions about the terrain other than the gridded elevation data, it is not possible to determine visibility from the observer to the surrounding points in less than  $\Theta(n_R^2)$  time because each of the surrounding points must be examined at least once and there are  $\Theta(n_R^2)$  such points.

## 4.2 Algorithm Considerations

The R3 algorithm meets the Criterion for Adequacy and Criterion for Appropriateness given in Section 1.4.1 but its cost of computation increases with the cube of the average number of points along a LOS. Figure 4.3 shows the operation of the R3 algorithm in the first octant. The algorithm evaluates a unique LOS from the observer location to each target location, as signified by the dashed arrows. In this case there are 10 target points, with the observer located at the lower left corner. Also indicated are instances of an *x-crossing* point and a *y-crossing* point.

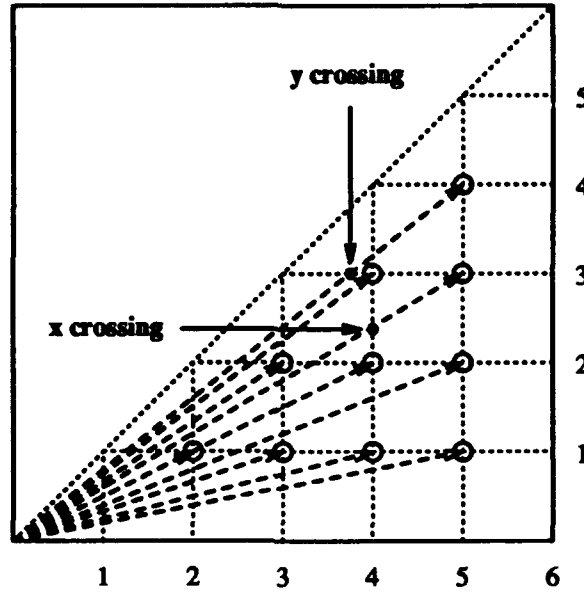


Figure 4.3: LOS Crossings of Grid Lines.

An x-crossing point occurs when a LOS intersects with a grid line in  $M$  which is perpendicular to the x-axis, and y-crossing points occur when a LOS intersects a grid line in  $M$  which is perpendicular to the y-axis. The observer location is not considered a crossing point.

In octants I, IV, V, and VIII there will be more x-crossing points than y-crossing points, with the reverse being true for the other four octants. Figure 4.3 depicts 10 LOS's (one for each target point), for which there are a total of 40 x-crossings and 20 y-crossings. In general, for a triangular first octant extending out  $x_g$  x grid lines from the observer point, there will be  $n = \sum_{i=1}^{x_g-1} i$  target points,  $\sum_{i=1}^{x_g-1} i(i+1)$  x-crossing points, and  $\sum_{i=1}^{x_g-1} i(i+1)/2$  y-crossing points. Note that the number of target points varies with  $x_g^2$ , that there are twice as many x-crossings as y-crossings, and the number of x-crossings and y-crossings varies with  $x_g^3$ . In octants I, IV, V, and VIII we observe that  $\Theta(n_R) \not\sim \Theta(x_g)$  and the R3 algorithm must make an elevation determination at every x-crossing. By symmetry, a similar relation holds for  $n_R$  and  $y_g$  in octants II, III, VI, and VII. Therefore, the R3

algorithm cannot run in less than  $\Theta(n_R^3)$

It would be desirable to construct an algorithm which ran closer to the intuitive limit of  $\Theta(n_R^2)$  than R3 does. If an approximation more simple than linear interpolation is made for the elevation and LOS height for all crossing points on a given segment (or given half-segment), computation would be reduced, but only by a constant scale factor. This would produce less exact visibility determinations, but may be acceptable depending on the uncertainty of the elevation data and the intended application. Several obvious approximations would be minimum, maximum, or simple average of the elevations and LOS-heights.

To achieve a running time faster than R3, an algorithm must either process LOS's to fewer target points ( $n_R^2$  for R3) or reduce the average cost of determining the LOS to a target point ( $n_R$  for R3). Figure 4.3 provides a visual clue to one aspect of the R3 algorithm that is generating the extra factor of  $n_R$  over our proposed minimum cost of computation. Especially on grid segments close to the observer, many elevation and LOS height determinations must be made for x-crossing points (other than grid intersections) which are only slightly apart. Note that the LOS height at a point, if accurately determined, completely represents all the necessary LOS information between that point and the observer that may be needed to determine the LOS height at points further out. A procedure to produce a suitable LOS height estimate in constant time for all LOS's crossing a given line segment would eliminate the multiple determinations on a single segment occurring in the R3 algorithm, and provide the desired running time of  $\Theta(n_R^2)$ .

This idea of consolidating elevation and LOS height information along a grid segment into an estimate represented at one or two points and using it as the basis for LOS determinations for crossing points on the next column (for the case of the first octant) of target points suggests the paradigm of an expanding *wave front* (see Figure 4.4). This has the apparent potential to improve running time over the

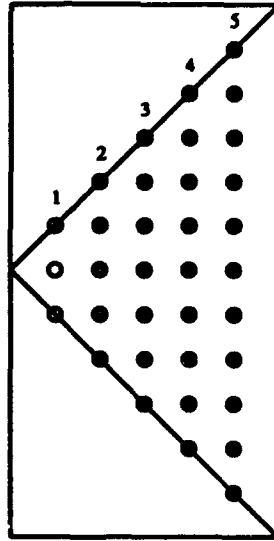


Figure 4.4: Progression of LOS Wave Front.

$\Theta(n_R^3)$  of R3 by reducing the cost of computing the LOS height of a target point to a constant (instead of  $n_R$  for R3). Several approaches have been developed to exploit this idea. [TD92] developed an algorithm, MB, which required  $\Theta(n_R)$  processors and  $\Theta(n_R)$  time. The fast LOS procedure for the Xdraw routine in [RFM92]) executes in a non-parallel environment in  $\Theta(n_R^2)$ . A similar implementation is described in [Bro80]. Unfortunately, all of these approaches suffer from what is referred to in [TD92] as *chunk distortion*. In the context used here, the Criterion for Appropriateness is violated because in the course of the progressive application of the approximation, elevation values that should have no influence on a given LOS propagate their influence as shown in Figure 4.5.

Consider first the triangle of points labeled  $a$ ,  $b$ , and  $c$ . For points in the first octant, the angle  $\theta$  formed by the LOS and the x-axis must be between 0 and  $\pi/4$ , so the LOS to a target point  $a$  must have an x-crossing on grid segment  $\overline{bc}$  that is not at  $b$  or  $c$ . Given the elevation data model being used, the elevation estimate of an x-crossing between  $b$  and  $c$  should be determined by the known elevation values

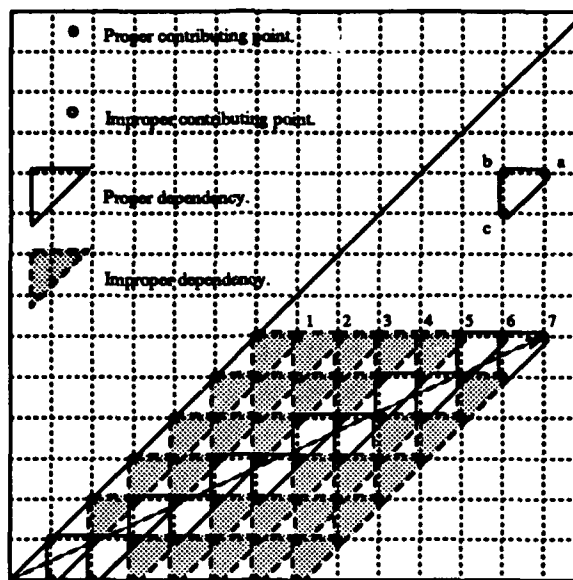


Figure 4.5: Erroneous Propagation in Wave Front Algorithms.

for  $b$  and  $c$ , so those points are *proper contributing points* in determining the LOS from the observer to  $a$ .

Applying the criteria for appropriateness and adequacy from Section 1.4.1 determines which points are proper contributing points in the determination of any given LOS. In Figure 4.5, the triangle encompassing  $a$ ,  $b$ , and  $c$  indicates that there is a *proper dependency* of the current LOS on all three enclosed points. A shaded triangle indicates the existence of an *improper dependency* — that one or more of the enclosed points are contributing to the determination of the current LOS but should not be.

With these terms in mind, the parallelogram region formed by the proper and improper dependency triangles encompassing a LOS from the observer to point 7 illustrates the problem that arises from straightforward wave front algorithms. One of the values properly contributing to the LOS determination for point 7 is the LOS height at point 6. This was calculated when the wave front algorithm was

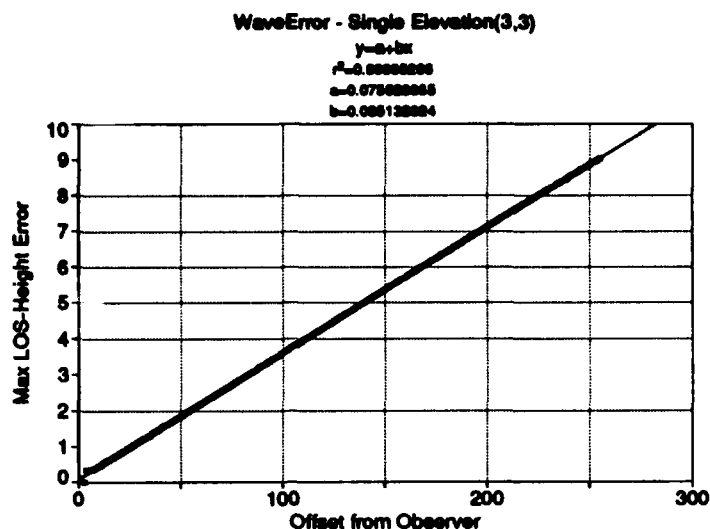
determining results for points along the  $x$  grid line containing point 6, and one of the points influencing the value calculated for point 6 was point 5. Looking one column further back in the progression of the wave front reveals the improper influence of point 4 in determining the LOS for point 7. Point 4 properly contributes to the LOS determination for point 5, but by attempting to capitalize on point 5's value (to avoid operations on values on grid lines closer to the observer and already passed by the wave front) to determine a LOS height value for point 6 and subsequently point 7, point 4 becomes an improper contributing point.

As suggested by the arrangement of contributing points and dependency triangles, the number of proper contributing points  $n_{PC}$  will vary from one to two times the number of  $x$  grid lines crossed by the LOS, depending on  $\theta$ . However, there are drastic fluctuations in the number of improper contributing points,  $n_{IC}$ . Assume an observer at coordinate  $(0,0)$  and a LOS out to a grid point  $(x,y)$ . Let  $\theta' = \arctan \frac{1}{2}$ . For  $\theta \rightarrow 0$  or  $\theta \rightarrow \pi/4$ ,  $n_{PC} < 2x$  and  $n_{IC} \rightarrow 0$ . For  $0 < \theta < \pi/4$ ,  $n_{PC} < 2x$  but as  $\theta \rightarrow \theta'$  then  $n_{IC} \rightarrow \frac{x^2}{4} - n_{PC}$ . In general, for  $0 < \theta < \pi/4$  then  $n_{IC} = O(x^2)$  as  $x \rightarrow \infty$ . This does not bode well for the wave front approach, since the ratio of improper contributing points to proper contributing points along a LOS with a given  $\theta$  will increase proportionally with  $x$ .

Given this result, the likelihood that one or more relatively extreme elevation values (such as a hill top or ridge line) will occur in the set of improper contributing points increases as the LOS length increases, although its impact decreases with its distance from the observer. A single such extreme value could produce patently erroneous LOS height results for a large fraction of the total target points, raising serious questions on the usefulness of a direct wave front approach.

Figure 4.6 shows the propagation of LOS-height error in an elevation dataset in which all elevation data values are zero except for the point  $(3,3)$ , which has an elevation of one unit. The horizontal axis shows distance from the observer in  $x$ -axis





**Figure 4.6: Propagation of Maximum Chunk Distortion LOS-Height Error with Distance from the Observer.**

intervals, and the vertical axis shows the maximum LOS-height error (based on the criteria from Section 1.4.1) at that x-axis distance. Note that the maximum absolute error increases essentially in a linear fashion with distance from the observer. Errors introduced relatively close to the observer, even if small, can produce large changes in LOS-height at longer ranges. Since 75% of the points in a circular region around an observer will be found at the longer (greater than or equal to  $R/2$ ) ranges, avoiding chunk distortion error is important for any accurate visibility algorithm.

It is possible to modify the wave front approach to reduce or eliminate the influence of improper points. [TD92] developed a modified algorithm, PB, which runs in  $\Theta(n_R)$  time but requires  $\Theta(n_R^2)$  processors and so leads to an  $\Theta(n_R^3)$  procedure for sequential machines. Modifying the Xdraw procedure in [RFM92] to incorporate careful control of the size and order of evaluation of subintervals of the wave front can eliminate either of the 'wings' of improper contributing points, but not both.

### 4.3 R2 Algorithm Development

We turn now to Figure 4.7, another representation of selected aspects of the R3 algorithm in operation in octant I, for insights into developing a more efficient way of determining or adequately approximating visibility from an observer to all grid intersection target points within a convex region. Target points on x grid lines 3, 4, and 5 are labeled *a* through *i*. The non-target point x-crossings of x grid lines 3 and 4 by LOS's to target points *a* through *d* are emphasized with small shaded square markers; x-crossings on lines 1 and 2 are not emphasized due to congestion.

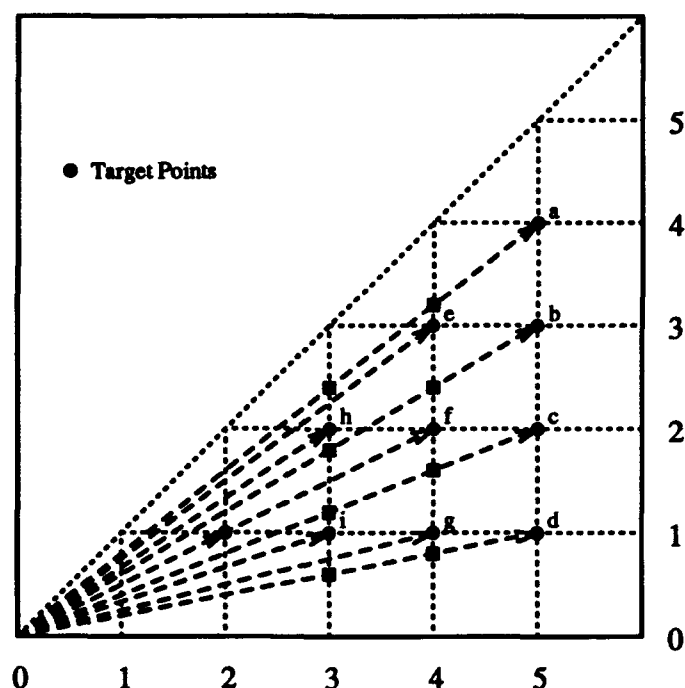


Figure 4.7: Operation of the Basic R3 Algorithm.

Since in the R3 algorithm the order in which LOS's to target points are evaluated is not important, let us presume that LOS's to points *a* through *d* are determined before any other points in the first octant. As a LOS is run from the observer to each of points *a* through *d*, it encounters 4 x-crossings. Let  $k_i$  represent

the x-crossing where a LOS from the observer to point  $k$  crosses x grid line  $i$ . While we know the visibility and LOS height at points  $a$  through  $d$ , we do not have any exact corresponding information about the first octant target points on x grid lines 1 through 4 (such as points  $e$  through  $i$ ). However, if during the operation of the algorithm we save the LOS height values at x-crossings, we do have potential estimates for the points  $e$ ,  $f$ , and  $g$  in the form of  $a_4$ ,  $b_4$ ,  $c_4$ , and  $d_4$  (the four shaded boxes on x grid line 4). This suggests a general approach of running LOS's to the outermost points of the region and using their intermediate results as estimates for the visibility and LOS height information for interior target points. We achieve a faster running time in comparison with R3 by reducing the number of points to which complete LOS's are calculated, instead of reducing the cost of running (or extending) a LOS to a specific point as in the wave front approach.

By visual inspection we can see that, of the x-crossings available,  $a_4$  is the best (in terms of spatial proximity) estimate for point  $e$  and  $d_4$  is the best estimate for point  $g$ . While the best estimate for  $f$ ,  $b_4$  or  $c_4$ , is not readily apparent to the unaided eye, it is an inexpensive determination to make during computation. Considering x grid line 3, we see that if we used R3 there would be 7 x-crossings, only the four emphasized crossings  $a_3$ ,  $b_3$ ,  $c_3$ , and  $d_3$  would be available here as estimates for points  $h$  and  $i$  because  $e_3$ ,  $f_3$ , and  $g_3$  would not have been computed. LOS's would not be run from the observer to points  $e$ ,  $f$ ,  $g$ ; their visibility and LOS height information would be estimated as described above.

Note that as we work our way back towards the observer, the x-crossing points on LOS's to the outermost points become more closely spaced and can generally provide estimates closer to target points than at x grid lines further from the observer.

An algorithm based on this idea of estimating interior point values from x-crossings of LOS's to 'boundary' points would calculate a number of LOS's proportional to  $2\pi n_R$ . The number of locations along each LOS at which operations (each

taking no more than constant time) would be performed would be proportional to  $n_R$ . Therefore, the expected running time of this method would be  $\Theta(2\pi n_R^2) = \Theta(n_R^2)$ . Since this is our targeted improvement over the  $\Theta(n_R^3)$  time in the R3 algorithm, we refer to this approach for now as the initial R2 algorithm.

To implement this algorithm it is necessary to define more formally what 'boundary' points are. Initially, let  $B$  be the set of points on the *boundary* of a convex region containing an observer such that all points in  $B$  are target points (and therefore the distance between them and the observer is less than or equal to  $R$ ) and they are 4-neighbor adjacent to at least one point that is not a target point (not in the convex region). An example of such points, labeled  $a$  through  $g$ , for the first octant is shown in Figure 4.8

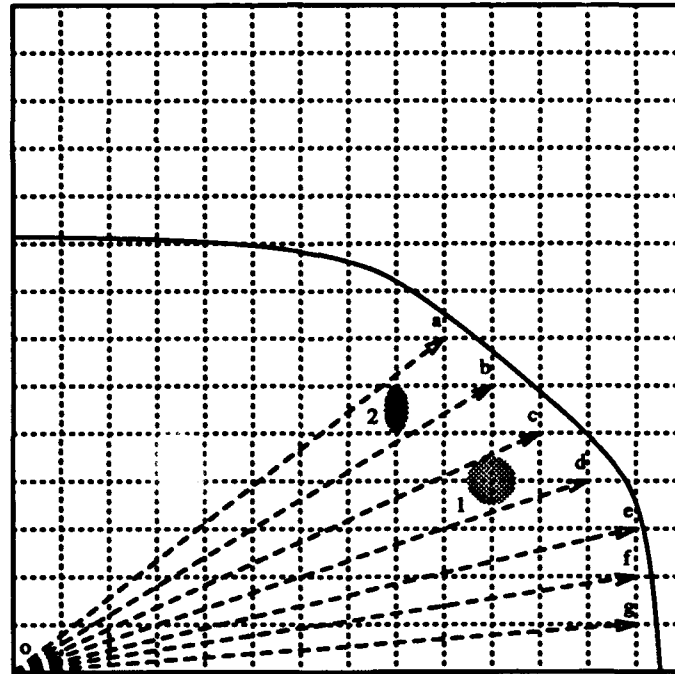


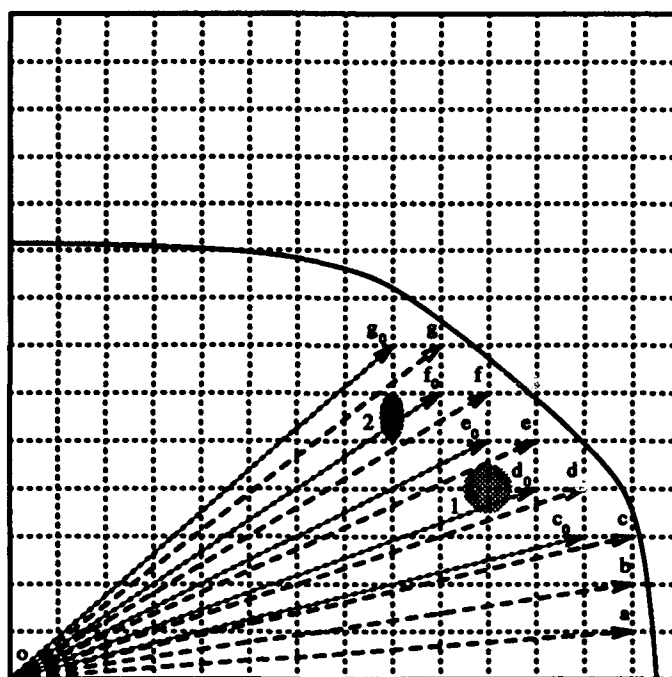
Figure 4.8: Coverage Problems in the Initial R2 Algorithm.

The convex but non-circular outline of the region of target points in Figure 4.8 illustrates two problems that can arise with the boundary definition for the initial

R2 algorithm. A type 1 problem is illustrated by shaded region 1. Intuitively, we desire that any crossing point that is to provide a LOS height and visibility estimate for a target point be within one-half of a grid segment length from that point. The grid intersection point at the center of shaded region 1 has no x-crossing point that close. A type 2 problem is illustrated by shaded region 2. In programming an implementation of the R2 algorithm, it is possible to make the code simpler and more efficient when (again from the perspective of the first octant) it can be assumed that the grid segment beneath every target point contains at least one x-crossing. This is perhaps less serious than a type 1 problem, but it would be useful if any modification to the initial R2 algorithm to prevent occurrences of type 1 problems also eliminated type 2 problems.

Both type 1 and type 2 problems derive from relatively wide angular separations between adjacent LOS's, which are in turn related to the distance between adjacent boundary points. These distances are greatest when  $\theta$  approaches  $\pi/4$  and the adjacent boundary points are not 4-neighbor adjacent (i.e., a line segment between them is a diagonal, not a grid segment).

Figure 4.9 illustrates a modification to the R2 algorithm that increases total computation by no more than a factor of two and which prevents occurrences of type 1 and type 2 problems. Here, the definition for  $B$  is changed so that all boundary points may be 8-neighbor (instead of 4-neighbor) adjacent to at least one point outside the region of target points. LOS's running from the observer to points  $c_0$ ,  $d_0$ ,  $e_0$ ,  $f_0$ , and  $g_0$  have been added as a result of changing the definition of  $B$ . In particular, the LOS to  $d_0$  removes the type 1 problem and the LOS to  $f_0$  removes the type 2 problem. The additional LOS rays also provide more spatially proximate estimates for many other interior points.



**Figure 4.9: Modified R2 Algorithm.**

#### 4.4 Validity of R2 Algorithm

The degree of agreement between the R2 and R3 algorithms can be evaluated in several ways, depending on the intended use of the visibility results. One direct way is to measure the difference in the LOS-height at each target point between the value calculated by R3 and the estimate provided by R2 and call this difference the R2 LOS-height error.

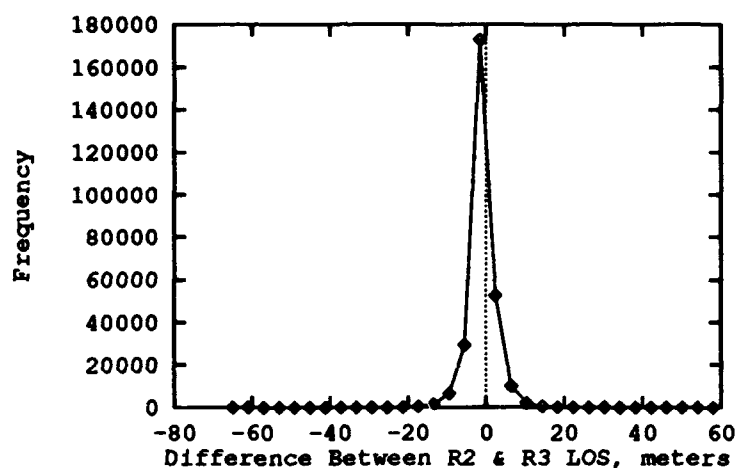


Figure 4.10: Distribution of R2 LOS-Height Differences (meters).

The degree of agreement between the R2 and R3 algorithms can be characterized by measuring the differences in the LOS-heights calculated by each method at each target point. A histogram of these LOS-height differences, computed using an observer-point with a high visibility index in DTED cell N37E127, is shown in Figure 4.10. In this figure the LOS-height produced by the R3 algorithm is taken as the reference height, with the R2 algorithm producing the estimated height. The difference between these two LOS-heights at a given point is the LOS-height error of the R2 algorithm.

Figure 4.10 shows a histogram representation of the distribution of these errors

over a circular area of ground radius 24,500 meters in the southwest quadrant of N37E127, encompassing over 276,000 interior octant points. The mean difference between calculated LOS heights was less than one meter, with a standard deviation of less than four meters. The basic statistics of the LOS-height error for these points are given in Table 4.1. Further confirmation of the validity of using the results from the R2 algorithm as an accurate visibility measure is presented in Figure 4.12.

LOS-height Error Statistics		
Mean error	-0.82	meters
Min error	-67	meters
Max error	60	meters
Std deviation	3.4	meters

**Table 4.1: Properties of R2 LOS-Height Error.**

Both Figure 4.10 and Table 4.1 suggest a generally high degree of agreement between the LOS-heights produced by each algorithm. Further evidence of the utility of the estimates produced by the R2 algorithm is provided by considering a simpler question: is or is not a target at a given point visible from the observer, without regard to specific LOS-height? Table 4.2 shows for the same sample area mentioned above the total interior quadrant points considered, the number of points where the two algorithms agreed on whether or not the point was visible from the observer, and the total number of points predicted to be visible by each algorithm.

Total Points	Matching Points	Visible by R2	Visible by R3
276,460	273,144	124,161	122,627

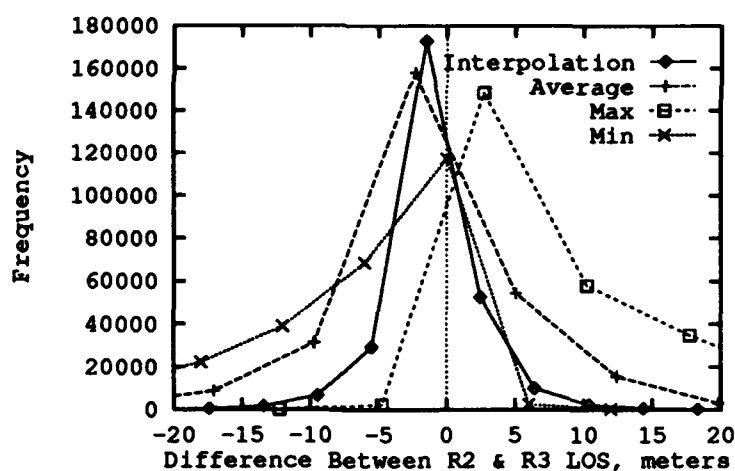
**Table 4.2: Count of Visible Points by R2 and R3.**

The two algorithms agree on a point-by-point basis on over 98.8% of points, and on their respective counts of visible points they vary by less than 0.6% of the total points. In the visibility results over two test sites from 7 GIS systems presented in [Fis93], the reported number of visible points within the potential viewsheds



varied from 1780 points to 2610 points for one site and from 1465 points to 2270 points for the other. The standard deviation of the reported number of visible points was 322.4 for the first site and 363.9 for the second. Compared with the reported variation in results from established GIS systems, the correspondence between R2 and R3 is excellent.

The preceding analysis was conducted using a version of the R2 algorithm that linearly interpolates elevations between 4-neighbor adjacent grid points as described in the model of the elevation data in Chapter 3. Some constant time savings can be achieved by employing other methods of estimating such elevation values, such as taking the maximum, minimum, or simple average of the two known elevations at the end of the pertinent grid segment.

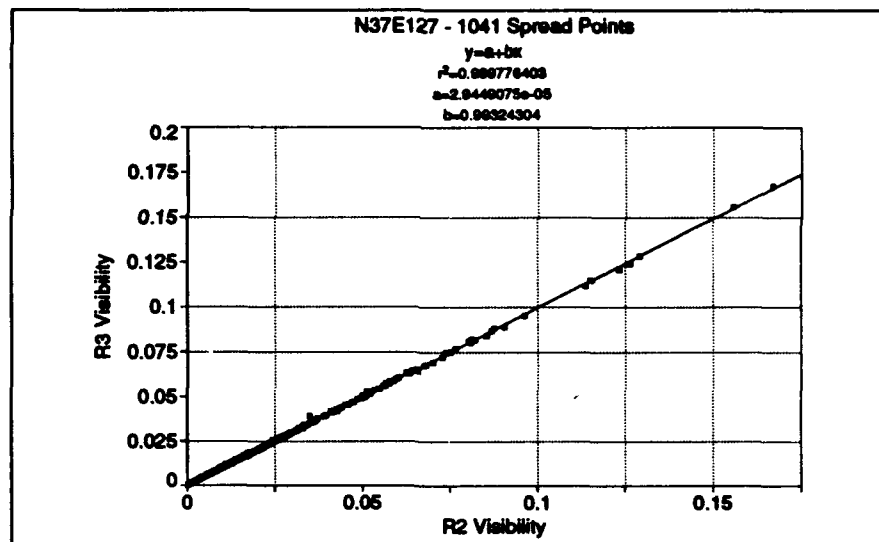


**Figure 4.11: Comparing 4 Variants of R2**

The distribution of LOS-height errors resulting from these and from linear interpolation are shown in Figure 4.11. While all produce good agreement with R3, the original linear interpolation method provides the best match in terms of minimizing the average and standard deviation of the absolute error. The cost for

its additional floating point interpolation operations is minimal compared to the basic cost of the operation of the algorithm, especially on a superscalar or other pipelined architecture.

A more systematic evaluation of the consistency of the agreement between the visibility fractions calculated by R3 and R2 was conducted by selecting 1041 points spatially distributed across the DTED cell N37E127. Figure 4.12 shows the high degree of agreement between the R2 and R3 results across the spectrum of visibility values for the points sampled. Each potential viewshed consists of 740,843 points. Approximately 25 days were required to compute the R3 values; R2 values were computed in less than 8 hours.



**Figure 4.12: Agreement of R2 and R3 Values for 1041 Points in N37E127**

Note that the form of the equation fitted to the observations is  $y = a + bx$ , with a very high 0.999776 degree of correlated variation between the R2 and R3 visibility fractions. The y-intercept value of 0.00002945 is very close to the expected value of 0, and the proportional term of 0.99324 is very close to the expected value of 1.

#### 4.5 Initial R2 Execution Time Results

The entire purpose of devising the R2 algorithm is to create a procedure with acceptable, if approximate, results with a running time of  $\Theta(n_R^2)$  instead of the  $\Theta(n_R^3)$  time for R3. Although it is clear from the design of R2 that it should achieve this goal, actual execution time information is in order for confirmation purposes.

Radius (meters)	Number of Points	R2 Time (seconds)	R3 Time (seconds)
24500	278K	18	896
12500	70K	4-5	131

**Table 4.3: Sample Execution Times of R2 and R3.**

Table 4.3 illustrates the variation of execution time with radius of interest for the same region in N37E127 used for Figures 4.10 and 4.11. As the radius of visibility increases by a factor of 1.96, the execution time of R2 increases by about a factor of 4. The time of R3 increases by close to a factor of 6.8, which is close to the factor of 7.5 predicted by a cube relation. Variations between expected and actual timings for different values of  $R$  are explained in part by the fact that the cost of operations at each crossing point will be different for visible and non-visible points. Visible points may require updates to the determining LOS data, whereas non-visible points do not. Since the ratio of visible to non-visible points for areas with different values of  $R$  will not in general be identical, their relative per-point execution times will vary within a bounded constant factor from direct quadratic or cubic relations. This is in addition to small variations due to virtual memory operations, caching patterns, etc., on the host workstation.

It is worth noting that even for relatively modest distances, Table 4.3 shows the R2 algorithm is already between one and two orders of magnitude faster than

R3 (similar program coding techniques having been employed in implementing each method). The performance advantage of R2 over R3 can be expected to increase linearly with the increase in the radius of visibility. An interesting speculation at this point is that the radius of visibility (for this region) at which R3 becomes more efficient than R2, due to the internal bookkeeping and boundary sequence control required in R2, may be at about 3000 meters, which is between 35 and 45 grid points for the sample dataset, depending on azimuthal direction. More detailed information on R2 execution time performance is presented in Chapter 7.

#### 4.6 Summary

Have our goals for the R2 algorithm been achieved? The modified R2 algorithm has run time  $\Theta(n_R^2)$ , and is therefore within a constant factor of optimality as discussed above. It provides exact results only for points in  $B$ , points on the boundary lines between octants, and for the relatively few interior points where a LOS to a point in  $B$  crosses a grid intersection point. For the majority of points for which only estimates are provided (in the form of exact results from spatially close crossing points), values for target points closer to the observer will on average have closer approximating points than target points further out towards the boundary. Figure 4.12 provides further evidence of the accuracy of R2 based on a large number of test points.

In a narrow sense the criteria for appropriateness and adequacy will not be satisfied in all cases, because the LOS from the observer to the estimating crossing point will be 'near' but not identical to a LOS from the observer to the target point. In a broader sense, however, these criteria are satisfied in that the problems with wave front algorithms illustrated in Figure 4.5 — the undue influence of extreme elevation points over large areas — have been eliminated.

Depending on the use intended for the results of the R2 algorithm, it may

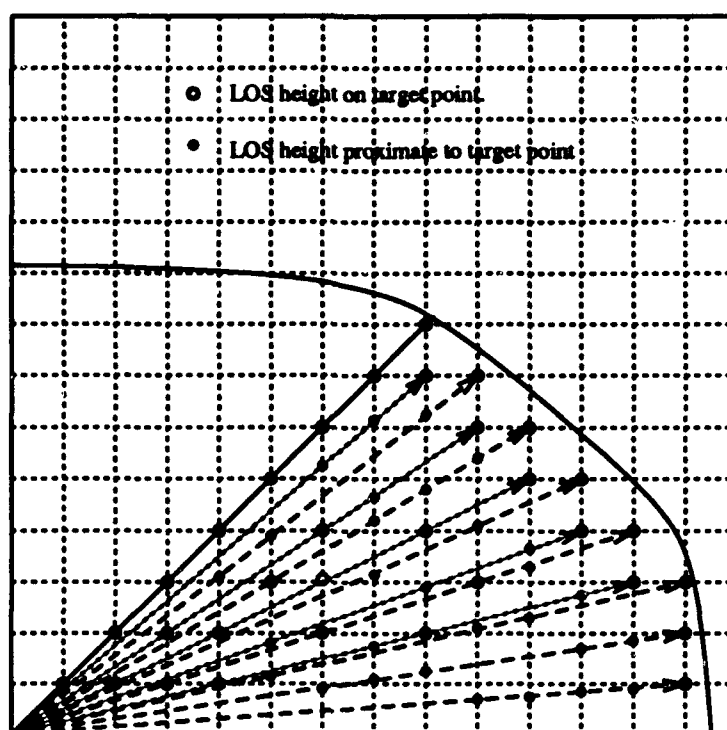


Figure 4.13: Alternative Tessellation of R2 Observations.

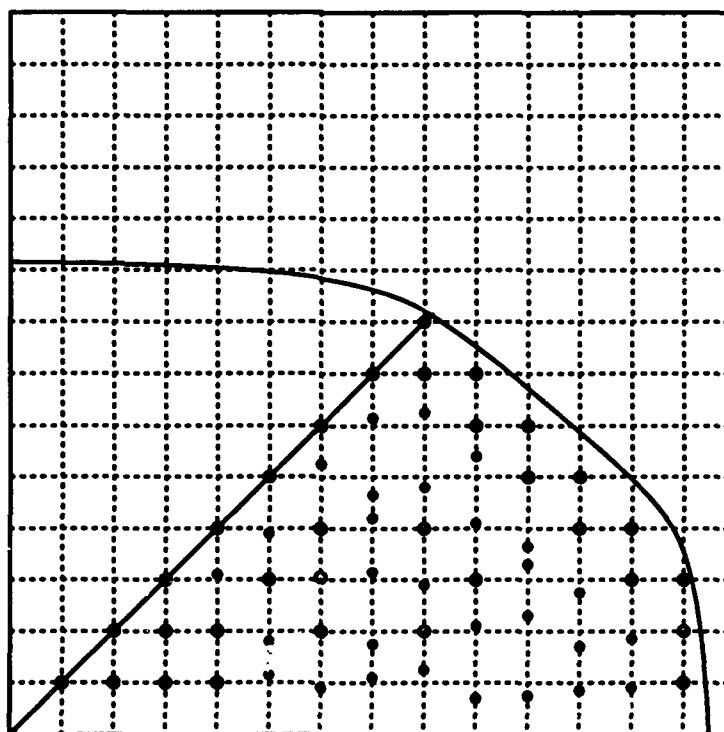


Figure 4.14: De-Cluttered Alternative Tessellation from R3.

be appropriate to view its operation as generating an alternative tessellation to one that corresponds to a normal rectangular grid. Figure 4.13 shows the pattern generated by the x-crossings of the modified R2 algorithm. For points in  $B$ , on the boundary between octants I and II, and certain interior locations, the points in this alternative tessellation correspond to those of the original elevation grid. Figure 4.14 shows the same points with the LOS lines removed for greater clarity. The increasing divergence between the elevation data tessellation and the R2 tessellation can be noted as distance from the observer increases.

Given that the original layout of elevation values is only one of many choices of patterns to represent the underlying terrain, it may be valid to consider the points produced by the R2 algorithm as an alternative pattern for representative visibility values, which closely follows but is not identical to the pattern of the elevation data. In this sense, the pattern produced by the R2 algorithm are not estimates for points in the elevation data pattern, but simply an alternative spatial sampling. For latitudes where the inequality of north-south and east-west ground spacing among elevation sample points in DTED data results in a pronounced rectangular (non-square) grid, the variations in the R2 pattern from the original grid may on average provide a more uniform sampling of the terrain.

Tradeoffs between computational cost and fidelity to the original grid can be made by extending the value of  $R$  used in determining the boundary points by some constant scale factor  $C$ ; for example, let  $R' = CR$ . In determining boundary points we follow a boundary that is a distance  $R'$  from the observer, but when actually running a LOS we only extend it to points within distance  $R$  from the observer. The cost of execution is now  $\Theta((2\pi n_{R'}R) = \Theta(2\pi Cn_R^2) = \Theta(n_R^2)$ , which is within constant factor  $C$  of R2 when  $R' = R$ .

A further generalization of R2 would be to employ an angular separation between adjacent LOS's that is not determined by ending each LOS on the exact

location of a boundary point. Other criteria could be used, such as minimizing the number of LOS's to avoid type 1 and type 2 errors, or providing the most uniform coverage (best match to the original grid) for a given number of LOS's. Another question that could be investigated is determining the fraction of interior points for which R2 computes an exact LOS height as a function of  $R$  and convex region shape.

## CHAPTER 5

### DEVELOPMENT OF LOS RAY ESTIMATOR ALGORITHMS

In this chapter we will propose several visibility index determination algorithms, all based on sampling potential visibility from an observer by constructing a set of equally spaced LOS rays from an observer. The computational cost for all of these algorithms is  $\Theta(n_R)$ , compared with  $\Theta(n_R^3)$  for R3 and  $\Theta(n_R^2)$  for R2.

The analysis of visibility information for a significant real world problem area (in excess of 23 million observer points representing approximately  $150,000\text{km}^2$  of terrain on the Korean Peninsula) was conducted to develop, refine, and validate visibility index estimation algorithms. Elevation data was adjusted according to Equation 3.1 prior to conducting visibility calculations. Selected terrain regions have been analyzed to examine changes in visibility estimates produced by variations in approximation parameters such as ray density and point-distance weighting.

#### 5.1 VL Algorithm

The basic VL (Visibility-LOS) algorithm to calculate an estimated visibility index from a specific observer location is shown below. It borrows from the Object-Oriented notation from the C++ programming language and some additional syntax from the Ada programming language. The fundamental object, **grid**, contains the adjusted elevation raster and various descriptive and visibility state information accessed and manipulated by method routines.

The method **Define.Elevation.Profile** creates a one dimensional vector of elevation values for each location on the grid where the LOS ray crosses an x grid line in octants I, IV, V, and VIII and where it crosses a y grid line in octants II, III, VI, and VII. Where the LOS ray crosses a grid line segment that is not on a grid point, linear interpolation between the endpoint values of the segment crossed provides the



**# Basic VL algorithm.**

```

Ray_Gridpoints := 0;
Visible_Gridpoints := 0;
FOR i IN 1 ... Number_of_LOS_Rays
    grid.Define_Elevation_Profile(Observer, LOS_Ray_Endpoint[i]);
    Ray_Gridpoints += grid.Values_in_Profile;
    grid.Calculate_LOS_on_Profile();
    Visible_Gridpoints += grid.Visible_Profile_Points;
Viewshed_Fraction_Visible := Visible_Gridpoints / Ray_Gridpoints;

```

elevation value used. The method `Calculate_LOS_on_Profile` takes the vector of elevation values which represents the one dimensional visibility profile and determines the visibility of each point in the profile.

**# Calculate\_LOS\_on\_Profile method.**

```

Controlling_Slope := -∞;
Points_Visible := 1;
FOR i IN 1 ... Number_of_Profile_Points
    Current_Slope := (Profile_Elevation[i] + Target_Height - Observer_Elevation) / i;
    IF Current_Slope ≥ Controlling_Slope THEN
        This_Slope := (Profile_Elevation[i] - Observer_Elevation) / i;
        IF This_Slope ≥ Controlling_Slope THEN
            Controlling_Slope := This_Slope;
        Points_Visible += 1;

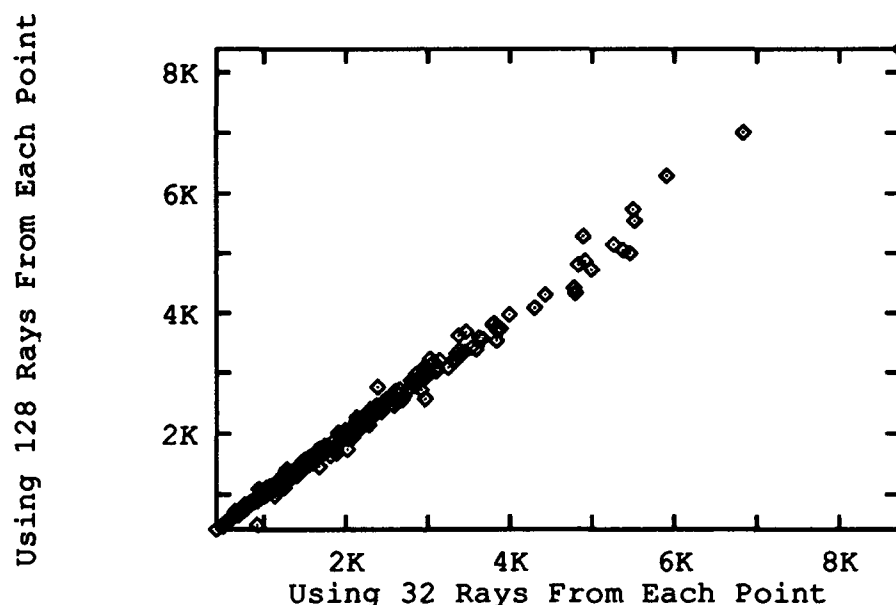
```

Note that the counter of the number of points visible from the observer is initialized to 1 instead of 0 because the observer point is always visible from itself.

## 5.2 Investigating Visibility Index Estimates

An estimate of the visibility index for each point in an area of interest can be computed by constructing some number  $n_{VI}$  of equally spaced LOS's from each observer out to the range of interest and summing the number of points visible

from the observer along each LOS. This sum can be scaled to make best use of the representation being used to store the result; for a 16-bit signed integer, the result could be scaled to the range 0–32767. Evidence that using LOS samples from each observer point can produce a consistent visibility index estimate was gathered by performing the same analysis using 32, 64, 128, and 256 LOS's from each observer point, using the VL algorithm. The results are shown graphically in Figure 5.1 by a sampling of 256 points from the terrain represented by cell N37E127, comparing results obtained using 32 rays and 128 rays.



**Figure 5.1: 40-kilometer Visibility Index Values, 32 and 128 LOS Rays, N37E127.**

Each point in the figure represents the visibility index values of the same location on the ground as estimated using 32 and 128 rays. The evident near-linear relationship of results obtained from each ray density suggests that using 32 rays produces estimates very similar to those achieved using four times as many rays;

even fewer rays may be appropriate for this particular cell. Also evident is the relatively small number of points with visibility values at the high end of the total range.

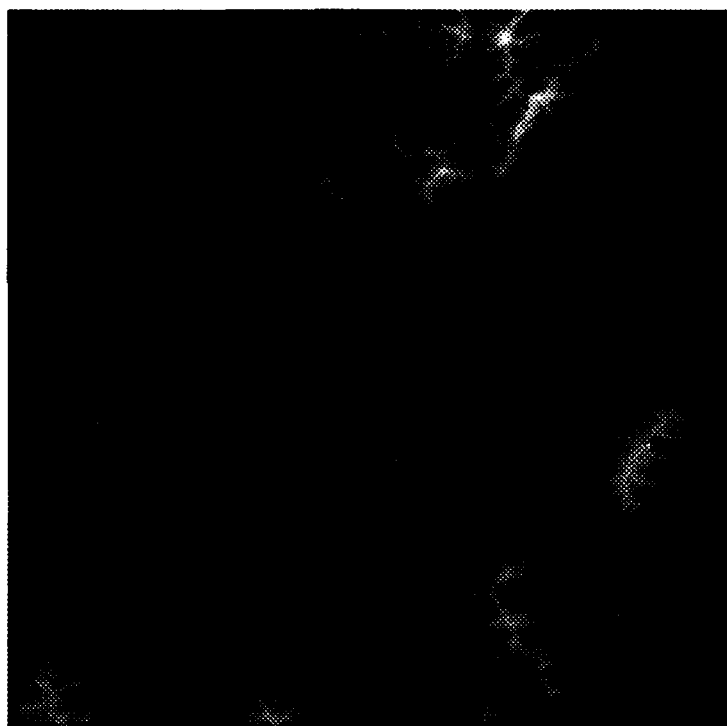
A gray-scale view of elevation in the northwest section of the DTED Level I cell N37E127 is shown in figure 5.2. This same area is shown in figure 5.3, with lighter levels of gray representing locations of high visibility. Initially, more than 50 days of execution time on a Sun IPC workstation were required to produce the visibility image for a DTED Level I cell. Subsequent refinements (see Section 5.4) reduced the time required to between 4 and 5 days. Parallel processing of separate terrain areas on networked workstations reduced elapsed time by a factor essentially linear in the number of workstations used.

### 5.3 Hypotheses and Characteristics

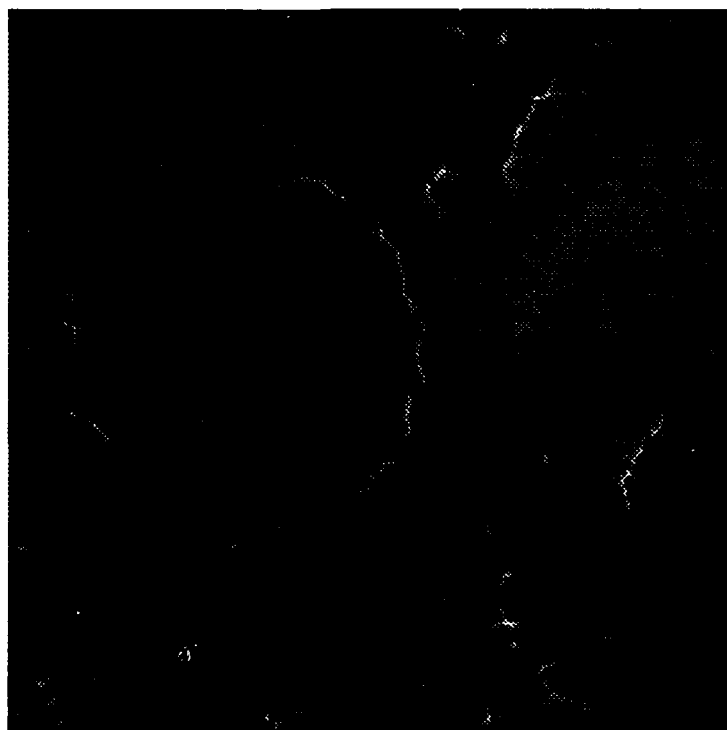
Consider the possible distributions of visibility values. If a very small number of points occur that have visibility indexes in the high range of visibility values for the entire AOI, this would significantly reduce the number of locations to be examined by specific site selection algorithms or interactive on-line analyses. Analysis conducted on over a dozen terrain cells suggests that such distributions are common on real terrain.

The density histogram of points having particular LOS values for the visibility indexes computed for N37E127 is shown in figure 5.4. The near linear decline in the log of the number of points as LOS value increases suggests a negative exponential relationship. For terrain where this or similar relationships occur, we can expect to find relatively few points with high LOS values.

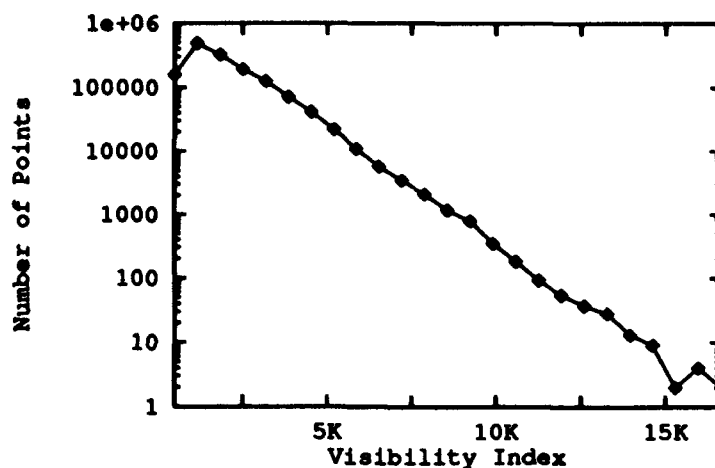
When spatial relationships among data points are ignored, an examination of corresponding elevation and visibility points in N37E127 reveals the lack of any obvious correlation between elevation and visibility values. Although the elevation



**Figure 5.2: Gray-Scale Elevation Image of DTED Cell N37E127.**



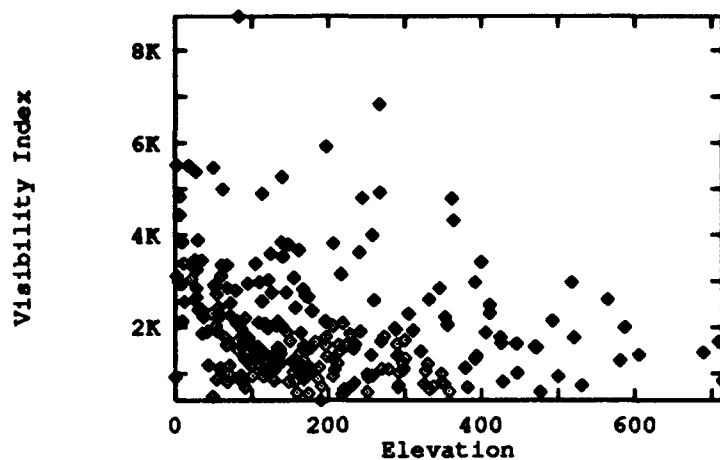
**Figure 5.3: Gray-Scale Visibility Image from DTED Cell N37E127.**



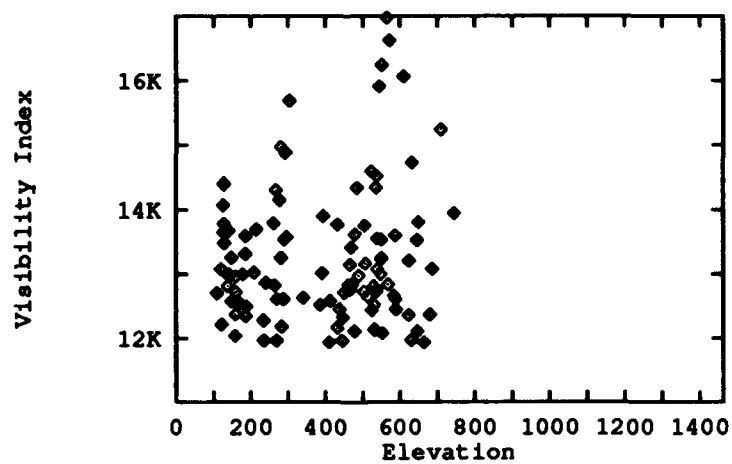
**Figure 5.4: Log Distribution of LOS Index Values from DTED Cell N37E127.**

values of data points have significant spatial correlations, the elevation of a given location is by itself only slightly correlated to its visibility index estimate. A sample of 256 points in figure 5.5 shows this graphically. Chapter 6 provides a more detailed examination of possible relationships between elevation and visibility.

Because it can be important in siting problems to identify a limited number of candidate sites with the *best* visibility (largest numbers of potential viewshed points visible), a legitimate question is whether some correlation exists between elevation and visibility for the best candidate sites, even if correlation is low overall. Evidence to date suggests that this is not the case. The pattern of elevation and visibility for the 110 points in N37E127 with the best VL visibility is shown in Figure 5.6. A more detailed presentation of the low correlation between the elevation and visibility index value at given points is provided in Chapter 6.



**Figure 5.5: Point-Matched Elevation and Visibility Values from DTED Cell N37E127.**



**Figure 5.6: Elevation and Visibility, 110 Best VL Visibility Points, N37E127.**

## 5.4 Pseudo-Fixed Point Optimizations

Although many programming languages lack built-in support for fixed point operations, careful partitioning of relative long (32-bit and greater) native integer representations can achieve the same effect, as long as attention is paid to the precision required by the problem and the number of times values must be converted to and from the pseudo-fixed point format. For example, since the maximum difference in elevation between any observer and target elevations on earth are known to be less than 16384 meters, only 13 bits in a 32-bit integer are needed to represent any such difference value. During visibility calculations where the divisor is an integer grid count, this allows a relatively accurate (18-bit fractional component of the quotient) pseudo-fixed point division to take place by shifting such an elevation difference value left (a computationally inexpensive operation on typical hardware) by 18 bits and performing a standard integer division. Use of such techniques to eliminate most floating point operations during the execution of the VL algorithm improved execution time by approximately a factor of 4 on Sun IPC hardware.

## 5.5 WeightF Algorithm

Real terrain manifests a significant degree of spatial correlation. We do not observe a mountain peak within one grid point of a valley bottom in normal elevation datasets. If we hypothesize that this correlation extends significantly to visibility subregions, then we could add to the accuracy of the visibility index estimates produced by the VL algorithm by using the visibility value for each point on a LOS ray to infer the visibility characteristics about the unsampled points in its vicinity. A straightforward way of doing this is to weight each point on a LOS ray by the percentage of the area within the total potential viewshed that is closer to it than to any other point on any LOS ray. This percentage is proportional to the distance of the point from the observer, which serves as the basis for modifying the VL

algorithm to produce the WeightF (Weighting-Full) algorithm.

**# Weight-F algorithm.**

```

Viewshed_Fraction_Visible := 0;
FOR i IN 1 ... Number_of_LOS_Rays
  grid.Define_Elevation_Profile(Observer, LOS_Ray_Endpoint[i]);
  grid.Calculate_Weighted_LOS_on_Profile();
  Viewshed_Fraction_Visible +=
    grid.Weighted_Points_Visible / grid.LOS_Ray_Area;
Viewshed_Fraction_Visible /= Number_of_LOS_Rays;

```

The individual LOS profile visibility calculation is modified to weight the count of visible points by the relative distance from the observer for each visible point. It also tracks a weighted count of all visible points which corresponds to the total area of the potential viewshed.

**# Modified LOS profile visibility calculation.**

```

Controlling_Slope := -∞;
Weighted_Points_Visible := 0;
LOS_Ray_Area := 0;
FOR i IN 1 ... Number_of_Profile_Points
  LOS_Ray_Area += i;
  Current_Slope := (Profile_Elevation[i] + Target_Height - Observer_Elevation) / i;
  IF Current_Slope ≥ Controlling_Slope THEN
    This_Slope := (Profile_Elevation[i] - Observer_Elevation) / i;
    IF This_Slope ≥ Controlling_Slope THEN
      Controlling_Slope := This_Slope;
    Weighted_Points_Visible += i;

```

## 5.6 Weight Algorithm

The standard implementation of the WeightF algorithm requires the accumulation of fractional contributions or other floating point representations to determine



the sum of contributions weighted by their true distance from the observer. An integer-only approximate weighting is employed in the Weight algorithm that uses the grid segment distance along the most rapidly varying axis as a proxy for distance from the observer.

**# Weight algorithm.**

```

Sum_Weighted_Points_Visible := 0;
Sum_Weighted_Total_Area := 0;
FOR i IN 1 . . . Number_of_LOS_Rays
    grid.Define_Elevation_Profile(Observer,LOS_Ray_Endpoint[i]);
    grid.Calculate_Weighted_LOS_on_Profile();
    Sum_Weighted_Points_Visible += grid.Weighted_Points_Visible;
    Sum_Weighted_Total_Area += grid.LOS_Ray_Area;
Viewshed_Fraction_Visible := Sum_Weighted_Points_Visible / Sum_Weighted_Total_Area;

```

This saves one floating point divide operation for each LOS ray computation and replaces a floating point addition with two integer additions. The impact of the resultant distortion in the weighting of points in LOS rays (off of the horizontal and vertical axes) is shown in Chapter 6.

## 5.7 Visibility Images

The VL, WeightF, and Weight algorithms have computational requirements that are low enough to allow the calculation of visibility index estimates over substantial viewsheds for large numbers of observer points. This offers the potential to create visibility images providing full AOI coverage that can be used for interactive exploration and analysis.

### 5.7.1 Magnitude-Only Visibility Images

By mapping visibility index values to intensity values for gray-scale display screens or even black-and-white hardcopy media with adequately fine dithering, a

visibility image of an area analogous to an elevation gray-scale image can be rendered (for example, see Figure 5.3). The ability to produce such images in quantity (i.e., over many areas and using many combinations of LOS parameters) offers a new way of displaying large amounts of visibility information in an intuitive fashion. The military applications of magnitude-only visibility images are obvious ([Ray94]), but many users of GIS capabilities may also find such visual, non-numerical representations appealing and useful.

### 5.7.2 Directional Visibility Images

Color displays have been used as an alternative to gray-scale representations for the purpose of depicting raster intensity values, such as elevation or visibility indexes. An alternative use of color capabilities is to map hues to directions, analogous to a color wheel. While this method does suffer from some limitations (such as the many-to-one mapping of visibility distributions to specific hues), it adds another way of visualizing visibility relationships in a region [FR94b].

## 5.8 Parallelism Considerations

Characteristics of many visibility algorithms are well suited to parallel execution at several degrees of granularity. During the conduct of this research, practical experience with parallelism on several levels was gathered. Even finer grained uses can be employed without significant changes to the basic algorithms. Operation of any of the VL-based LOS ray algorithms can be executed for the same observer by dividing the  $n$  total rays over  $m$  available processors (assuming  $m \leq n$ ), although the efficiency of such a scheme would be heavily dependent on memory organization and capacity. Even an interactive use of the R2 algorithm could easily parallelize over 8 processors by assigning one octant per processor.

- *WAN-Connected Stations.* Workstations geographically dispersed (at Troy, New York and at West Point, New York) executed visibility analysis procedures on locally stored copies of elevation and output datasets. Summary results and visibility datasets were collected via the Internet. One disadvantage was the need for separate copies of input datasets. The advantages included reliability (both sites were down intermittently but never at the same time), ease of parallelization, and lack of mutual interference. Monitoring was straightforward via multiple windowed telnet sessions and Network File System (NFS) file status monitoring and distribution. To fully exploit the overall reliability of geographically distributed computation, robust and automatic checkpoint and restart facilities were built into the software implementing the visibility algorithms.
- *LAN-Connected Stations.* At each site, multiple workstations shared common input datasets via LAN-connected shared file systems. A disadvantage was that outage of key file servers could affect all running analyses since the file system was not redundant. The ability to maintain single copies of input and intermediate files was an advantage. Problem sets as small as 30,000 observer points were divided across physical workstations, with control and results consolidated through shared directories.
- *Multi-Processor Stations.* Individual workstations, such as the Sun 670-MP with 2 to 4 physical processors each, were used to run visibility analysis on processors while leaving adequate processors free to service interactive loads. Although not exploited, physical memory and virtual memory demands could have been reduced through the use of shared memory segments.

## 5.9 Summary

In this chapter we have proposed three  $\Theta(n_R)$  visibility index estimation algorithms: VL, WeightF, and Weight. Investigation on real terrain showed consistent visibility estimates can be produced by the common underlying approach of sampling visibility by calculating a set of equally spaced LOS rays from an observer. The results of the visibility index analysis on a sample elevation dataset suggested that observer points with relatively high visibility index values may be a very small part of the total set of observer locations. Programming optimizations and use of parallel execution to speed up visibility analyses was also described.

## CHAPTER 6

### ACCURACY OF VISIBILITY ESTIMATES

This chapter investigates the critical issue of how accurately various  $O(R)$  visibility algorithms based on determining the visibility along a fixed number of LOS rays can estimate the true viewshed fraction from an observer. The R2 algorithm is used as an accurate reference value of true viewshed fraction because it requires significantly less computation than R3.

#### 6.1 Overview of Methodology

Unless otherwise stated, all visibility calculations in the following examples were calculated using a 40 km radius, a target height of 25 meters, an observer height of 5 meters, and were taken from the DTED Level I one-degree cell with a southwest corner located at 37 degrees north latitude, 127 degrees east longitude (N37E127). This cell contains elevation values ranging from 0 to 1462 meters above mean sea level. Mean elevation is 197 meters, median elevation is 152 meters, and the standard deviation of elevation values is 161.0 meters. Visibility measures are stated as the fraction of points in the potential viewshed at which a particular algorithm determined or estimated a target would be visible from the observer location.

The algorithms employed include:

- R2 The circumference sweep point-by-point visibility estimation algorithm described in Chapter 4.
- VL The initial visibility estimation algorithm employing a specified number of LOS rays without weighting individual points described in Chapter 5.
- WeightF A visibility algorithm employing a specified number of rays that uses an accurate area weighted technique, described in Chapter 5.

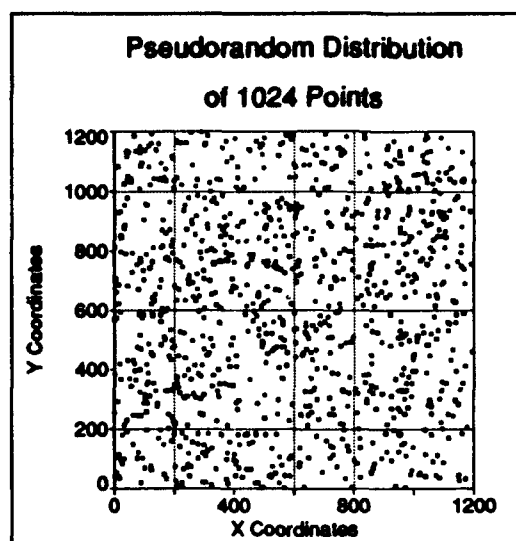
- Weight A visibility algorithm employing a specified number of rays that uses an approximate point weighting technique, described in Chapter 5.

A commercial curve-fitting and analysis software package was used in selecting and evaluating possible equations and parameters that could best describe the observations. The equations selected for display in the following figures were selected based on best correlated variation, strongest relationships suggested by F-statistics, elimination of spurious fits (step functions, sinusoidal functions, peak functions, etc.) on observations that lacked any strong fit, preference for simpler functions to more complex ones when the degree of fit provided was essentially the same, and elimination of equations that showed nonsensical behavior immediately outside the region for which observations had been collected. The equations relating visibility measures, or elevation and visibility, that were most often selected include:

- $y = a + bx$
- $y = a + bx^2$
- $y = a + bx^3$
- $y = a + b \log x$

It is of interest to note that using the criteria mentioned above, equations with multiple  $x$  terms such as  $y = a + bx + cx^2$  were generally not preferred to equations with only a single  $x$  term, such as  $y = a + bx^2$ . This may have occurred in many cases because equations with quadratic and higher  $x$  terms were selected to fit observations where all curve fits were weak and no underlying lower-order trend justified inclusion of other terms.

For many visibility and elevation samples used to analyze the various algorithms, a standard set of 1024 pseudorandomly distributed point coordinates was selected. The resulting pattern is shown in Figure 6.1. The  $x$  and  $y$  coordinate for



**Figure 6.1: Spatial Distribution of 1024 Pseudorandom Points.**

each point were drawn from separate pseudorandom number generators provided in the class libraries furnished with the Free Software Foundation's G++ compiler. Different seeds were used to start each generator. The generators permute the output of a Linear Congruential method [Knu81] with a Fibonacci Additive Congruential generator to obtain improved independence between sequential samples.

The R2 algorithm will be used as a reliable and computationally efficient reference to evaluate the accuracy of the visibility fraction estimates produced by other algorithms. Evidence of the agreement between the results from the R2 and R3 algorithms was presented in Chapter 4.

## **6.2 Performance of the VL Algorithm**

Figure 6.2 suggests that the VL algorithm has substantial capability to predict visibility (as measured by R2 visibility values on the Y axis). The correlated variation of 0.776 is evidence of a strong relationship, and the fitted equation  $y = a + bx^2$  is free from fitting artifacts and bizarre behavior at the extremes of the range of

the independent variable (the estimated visibility fraction calculated by the VL algorithm).

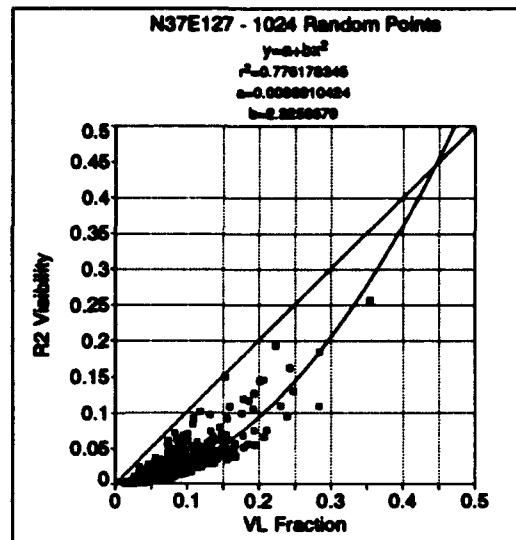


Figure 6.2: N37E127 - VL and R2 Visibility for 1024 Random Points.

Note that the estimates produced by the VL algorithm for most points substantially overestimate the visibility fraction calculated by R2. For example, points with a VL value of around 0.15 have R2 values clustered close to 0.05. It would be expected that points with very high actual visibility (e.g., centered on an open plain) would be accurately estimated by VL. We can observe that this hypothesis is supported in Figure 6.2 because points of very high VL estimated visibility (the right side of the graph) are as a percentage less overestimated than points near the center of the graph.

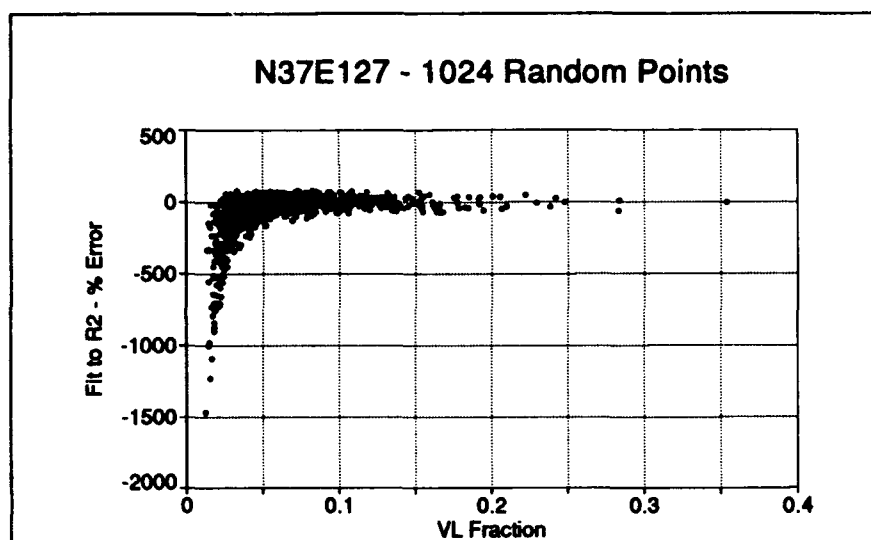
We might infer that relatively few points would be found beyond the VL values at which the fitted equation intersects  $y = x$ . This is because at and beyond that point the high degree of agreement between VL and R2 might indicate a boundary condition for the relation, just as a high degree of agreement occurs at the boundary of lowest visibility near (0,0). For the fitted equation in Figure 6.2 the intersection



between it and the  $y = x$  line would occur for a visibility fraction of about 0.4455 for VL and R2, and in fact in this figure and in Figure 6.4 (which focuses on high visibility points) very few points are found near or beyond that value.

The relative uncertainty in visibility associated with the VL estimated value decreases as the estimated visibility increases. This suggests that the VL algorithm, in spite of its poorer performance at lower values, might still be suitable for identifying the locations of highest visibility in an area. Sets of the 256 and 100 highest VL estimated visibility points will be used below to more closely examine visibility relationships present in high visibility points.

By examining the residual error after fitting  $y = a + bx^2$  to the points in Figure 6.2, we can gain a better understanding of the performance of the VL algorithm as a predictor of visibility over a range of visibility fraction values.

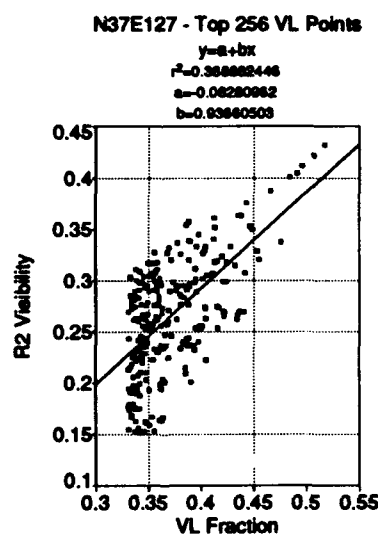


**Figure 6.3: N37E127 - Fit Residuals, VL and R2 Visibility, for 1024 Random Points.**

Figure 6.3 shows the percentage residual error after the curve fit to the VL and R2 values from Figure 6.2. Ignoring the large error percentages encountered for VL values close to 0 as manifestations of the small number problem, it can be observed

that using the estimates employed by VL in conjunction with a simple adjustment calculation (i.e., the fitted equation from Figure 6.3) produces estimates of visibility fractions whose relative error decreases as the visibility fraction increases. This is further evidence of the potential utility of the VL algorithm in finding points with the highest visibility.

Figure 6.4 suggests that for points of genuine high visibility, the VL algorithm provides better estimates (based on percent residual error) than for points of moderate to low visibility. Only points with the highest 256 VL estimated visibility values out of the total 1.44 million ( $1201^2$ ) points in the N37E127 cell are shown. Note that only the low computational cost of the VL algorithm (compared to the R2 or R3 algorithm) makes it practical to compute visibility estimates for every point in the cell, consuming approximately four days of computing time on a Sun IPC workstation.



**Figure 6.4: N37E127 - VL and R2 Visibility for Top 256 VL Points.**

Comparing the range of actual R2 visibility values in Figure 6.4 with those in Figure 6.2 demonstrates that points of high visibility have in fact been selected

by choosing points with the highest VL visibility estimates. For points with VL estimated visibility values above 0.4 there are no R2 values less than 0.2, and in all of the 1024 pseudorandomly selected points in Figure 6.2 only one point had an R2 visibility in excess of 0.2.

The layout of points in Figure 6.4 again demonstrates that the VL algorithm for many points underestimates the visibility fraction, although with decreasing severity as the visibility values increase. For VL values above 0.48 in this set of observations, underestimation seems to have disappeared or been subsumed in more random error from the straight linear regression of R2 on VL.

The ability to accurately predict which points will have high visibility values shows the suitability of the VL algorithm for many purposes that focus on points with the 'best' visibility.

### 6.3 Relationships to Elevation

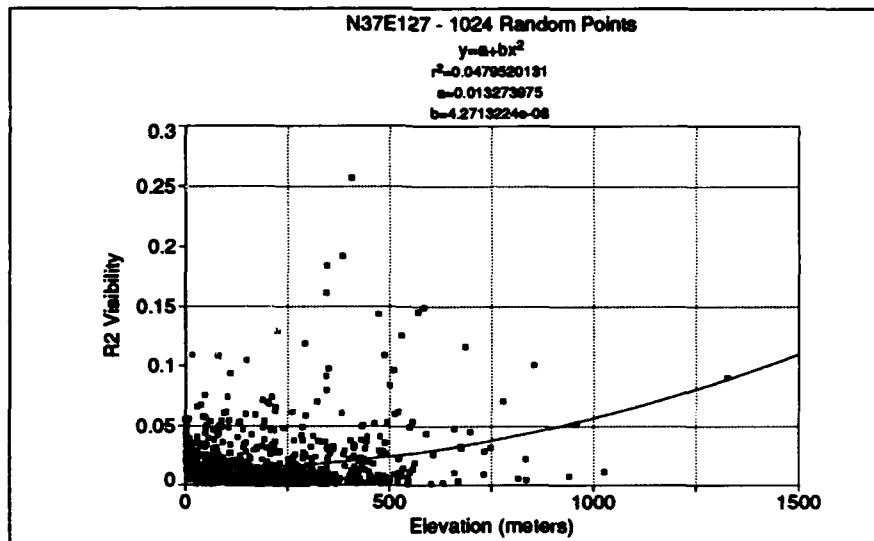
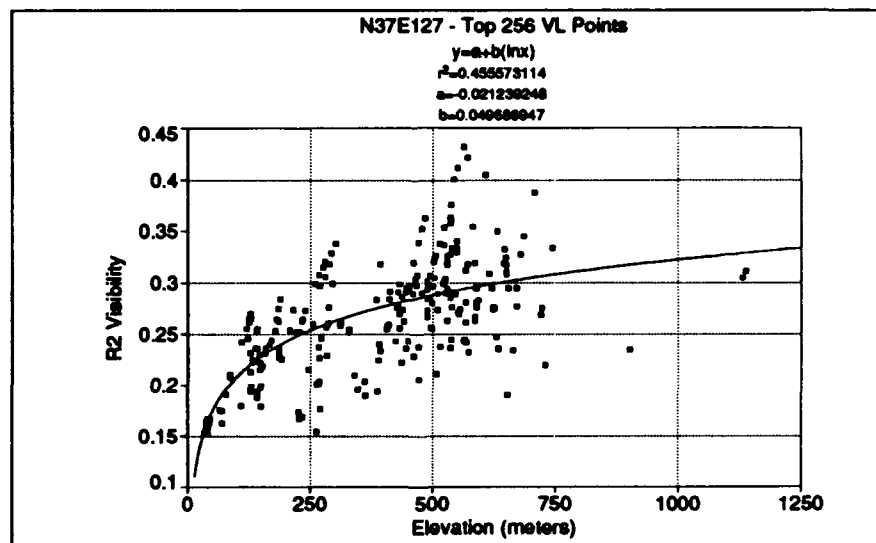


Figure 6.5: N37E127 - Elevation and R2 Visibility for 1024 Random Points.

Appendix C provides a detailed investigation of elevation and visibility relationships in different quadrants of the cell N37E127. To show an overview here of the relationships between elevation and visibility, a set of observations of elevation and R2 visibility fractions is shown in Figure 6.5. It plots, for the set of 1024 pseudorandomly selected points shown in Figure 6.1, the elevation in meters against the fraction of their total potential viewshed which was found visible by the R2 algorithm. The fitted equation,  $y = a + bx^2$ , might suggest a trend to a moderate increase in average visibility as the elevation of points increases. However, the scattering of points, low correlated variation (0.048), and presence of the highest visibility points in the middle of the elevation range argue that there is no basis for inferring a genuine, direct relationship between elevation and visibility.

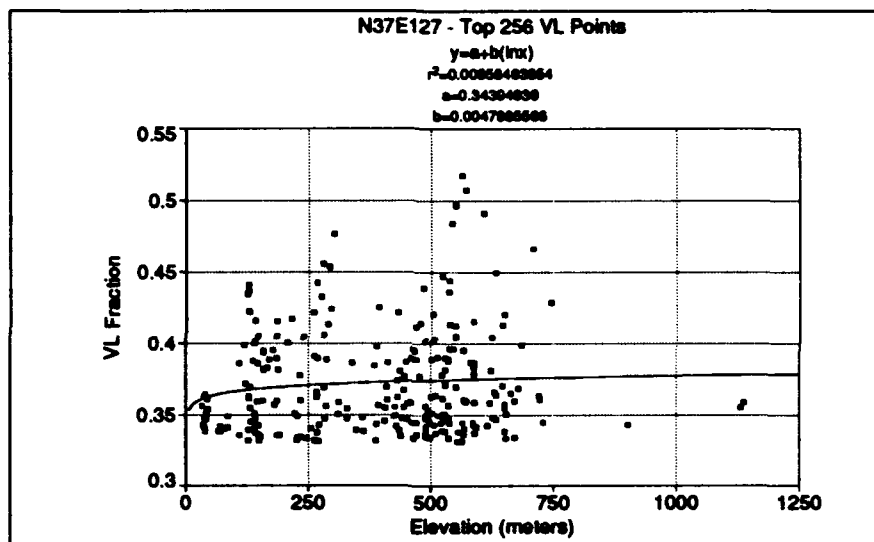
Figure 6.6 makes an initial use of the top 256 VL valued points from the N37E127 cell to examine possible relationships between elevation and visibility among points with high visibility values.



**Figure 6.6: N37E127 - Elevation and R2 Visibility for Top 256 VL Points.**

A moderate relationship ( $r^2 = 0.4556$ ) exists between elevation and visibility

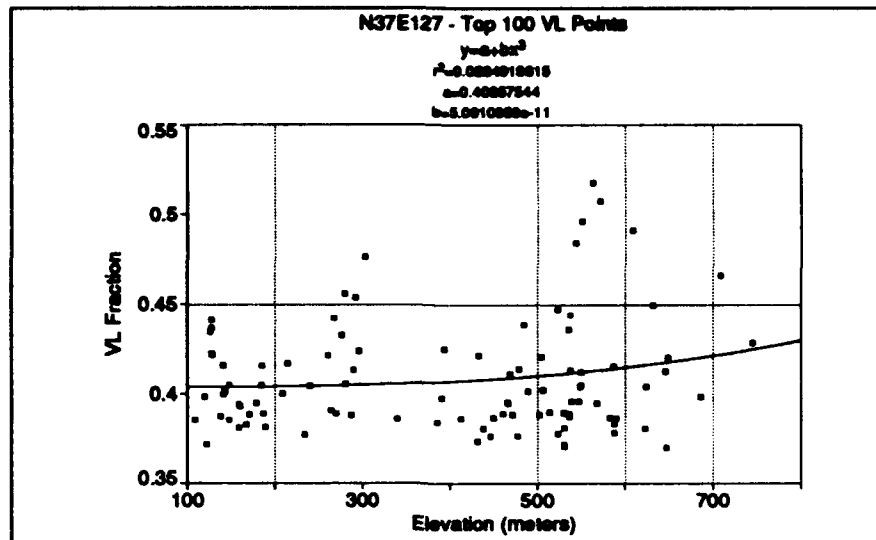
using the equation  $y = a + b \log x$  for this set of observations. However, recall that these points were *not* selected using their raw elevation values; they were selected using their estimated VL visibility fraction. What is shown in Figure 6.6 is a correlation between visibility fraction and elevation in a set of points known to have relatively high visibility. Even this correlation is not necessarily very useful. It only suggests a limited increase of visibility with elevation and provides no direct clues on how to use elevation values by themselves to select the highest visibility points, which in fact occur near the middle of the elevation range (around 550 meters).



**Figure 6.7: N37E127 - Elevation and VL Visibility for Top VL 256 Points.**

Figure 6.7 shows the virtually nonexistent relationship between elevation and the higher visibility estimates produced by the VL algorithm, with an extremely low correlated variation of 0.0096 using a fitting equation of the form  $y = a + b \log x$ . The virtual lack of correlation confirms the impression gathered by visual inspection, which is that raw elevation is not a suitable predictor of the visibility values estimated by the VL algorithm. As previously shown in Figure 6.4, in the high visibility ranges VL is a good predictor of the accurate visibility values produced by

R2.



**Figure 6.8: N37E127 - Elevation and VL Visibility for Top VL 100 Points.**

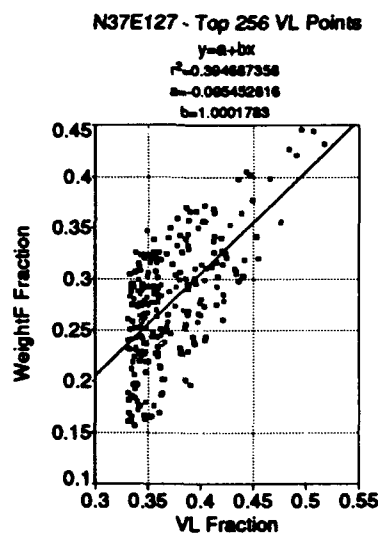
Figure 6.8 demonstrates that when restricted to an even more select set of the highest 100 estimated visibility points, there continues to be no substantial relationship between elevation and visibility. The actual correlation value, 0.0235, has increased slightly but remains quite small, and the form of the fitted equation has changed to  $y = a + bx^3$ . Note that the second derivative has changed from negative in Figure 6.7 to positive in Figure 6.8, suggesting further that even the best available curves represent spurious fits to data on measures (elevation and visibility) possessing essentially no direct relationship.

Whether over the general range of elevations and visibility values, or when focusing specifically on the best prospective points for high visibility, individual elevation values provide no consistent predictive ability for visibility. This observation holds true even when the points being sampled have been restricted to those predicted by a valid estimation algorithm to have high visibility.

#### 6.4 Improvements to VL Estimation Algorithm

As described in Chapter 5, with a relatively modest constant-time increase in the computational requirements of the VL algorithm we can attempt to make use of the spatial correlation of real terrain by weighting each point on a LOS ray in proportion to the area it 'represents' – the area to which it is the closest point from all points in all LOS estimating rays. This weighting will be proportional to the distance of the point from the observer. The Weight algorithm uses an integer-based approximate weighting of sample points by their grid distance from the observer along a LOS ray. A full weighting using floating point arithmetic is employed in the WeightF algorithm.

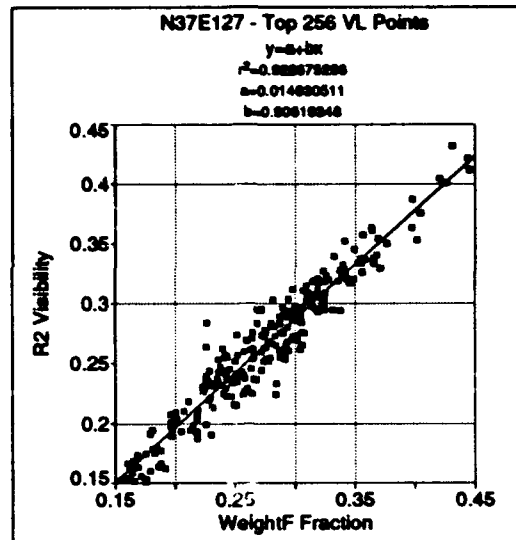
Figure 6.9 shows the impact on estimated visibility fractions of the previous set of 256 points with high estimated visibility of applying full weighting to the VL algorithm.



**Figure 6.9: N37E127 - VL and WeightF Visibility for Top VL 256 Points.**

Referring back to Figure 6.4 which depicted visibility results from the VL and R2 algorithms, we note a striking similarity. This suggests that when a definitive

comparison of the results of the WeightF algorithm is made against the standard benchmark results from the R2 calculations, there should be a strong correlation. This key result is confirmed in Figure 6.10.



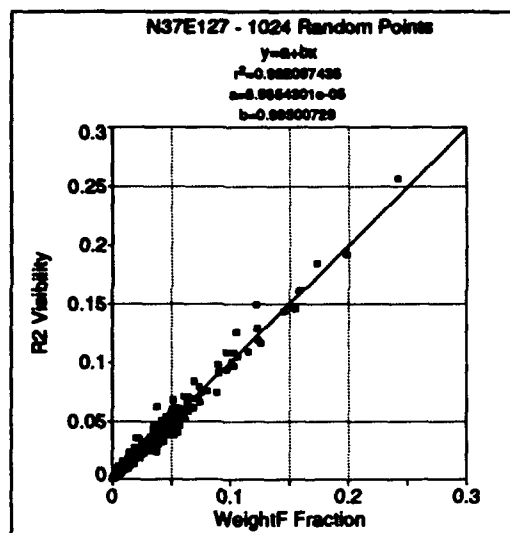
**Figure 6.10: N37E127 - WeightF and R2 Visibility for Top VL 256 Points.**

There is an impressive agreement over the entire range of visibility values. In addition to the high correlated variation of 0.9227 and the appealing linear form of the fitted equation, there is evidence that the overestimation problem from the VL algorithm has been significantly mitigated. WeightF is shown to have the potential to be an outstanding visibility fraction estimator.

Figure 6.11 shows the relationship between WeightF and R2 values as we broaden the focus from high visibility points to a range of pseudorandomly selected points. The correlated variation of 0.9821 is even higher than in Figure 6.10. Note that in this and some subsequent figures different (although still linear)  $x$  and  $y$  distance scales are used to better show detail in the data and curve fits.

The linear form of the fitted equation remains. The proportional term of 0.995 is very close to one, and the constant term in the equation (0.00008965) is very small

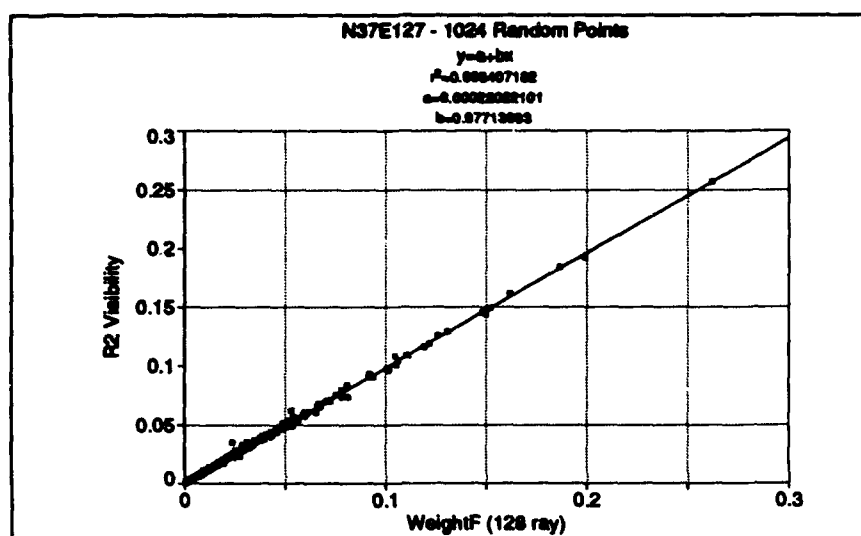




**Figure 6.11: N37E127 - WeightF and R2 Visibility for 1024 Random Points.**

in relation to the quantities being related. Together they suggest not only a good fit, but that there is no substantial over- or underestimation of visibility taken over the entire sample set. For the very high visibility points with WeightF and R2 visibility fractions above 0.15, the linear fit remains strong and all points are ordered correctly. In other words, if we rank all locations in order using the WeightF algorithm and then using the R2 algorithm, the ordering would be the same. In addition to its general robustness, this is further evidence of the utility of WeightF in identifying points with the best visibility.

Although a strong relationship to R2 is shown in Figure 6.11 when using WeightF with 32 LOS rays, this suggests the question of how much the correlation would improve with more rays. Continuing with the practice of using ray counts which are powers of 2 and mindful that many measures that are essentially descriptive statistical values improve not linearly but with the square or higher power of the sample size, an appropriate value for an increased ray count higher than 32 will be 128. Figure 6.12 shows the results obtained by running WeightF on the same

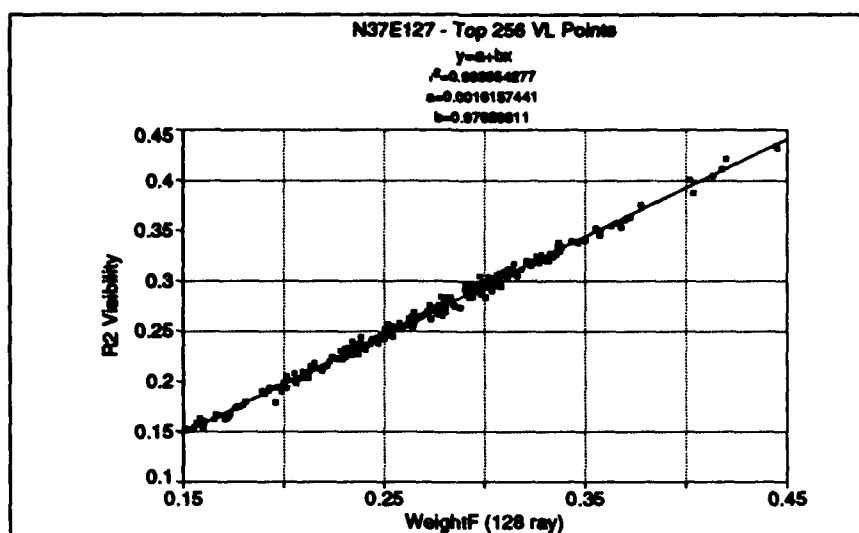


**Figure 6.12: N37E127 - WeightF (128 rays) and R2 Visibility for 1024 Random Points.**

points used in Figure 6.11, but using 128 rays.

The increase in correlation from an already high 0.9821 to 0.9984 is noticeable, but the utility of such an increase when compared to the increased computation cost (a factor of 4) will depend on the specific application and the inherent uncertainty in visibility values produced by errors presumed present in the data (see Section 6.8 below). The RMSE decreases from 0.0030486 for 32 rays to 0.00090769 using 128 rays, a decrease of approximately a factor of 3. Both in absolute and relative terms, the bulk of the error is concentrated in regions of low visibility.

A visual inspection of Figure 6.11 and Figure 6.12 suggests that the linear model fits particularly well at higher visibility values, a conclusion supported by an examination of the values of residuals between the observed data and the fitted equations. As shown in Figure 6.13, increasing the number of estimating rays from 32 to 128 for the set of high visibility points increased the correlation from 0.9227 to 0.9940, an increase which might be justified in trying to sort out the points with the very highest visibility. A valid strategy for finding points with the best visibility

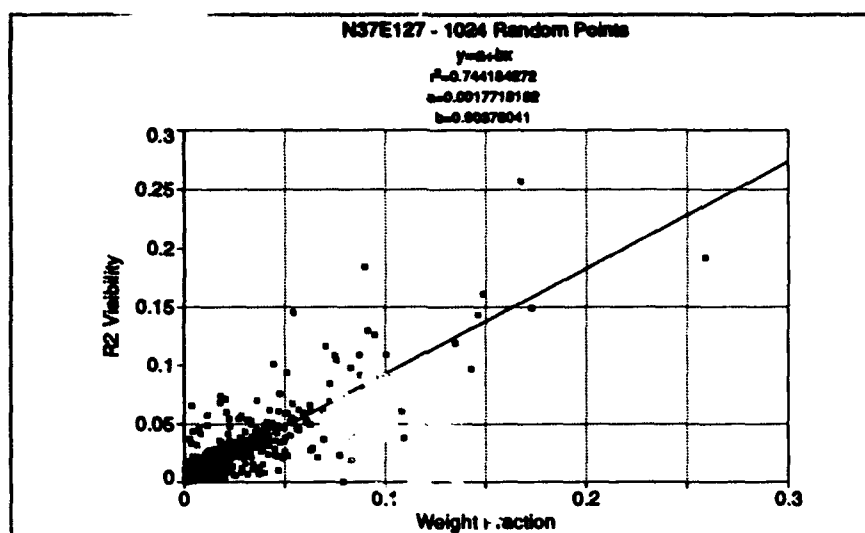


**Figure 6.13: N37E127 - WeightF (128 rays) and R2 Visibility for Top 256 VL Points.**

might be to run WeightF with a relatively modest ray count such as 32, selecting a set of candidate points based on this initial run, and then running WeightF with a higher ray count. This would only result in a constant time improvement, however, and so the utility of this approach will depend on the specific application.

The Weight algorithm runs faster than WeightF by a constant factor because it weights points by their grid distance from the observer along the most rapidly varying axis instead of tracking its fractional distance out along the LOS using floating point operations. We would expect that any systematic error introduced by this approximate weighting method will result in an overestimation of visibility. This is because for rays furthest off of either axis, the contributions of their furthest points will be undervalued. This will result in an increase in the significance of points closest to the observer, which tend to have a greater fraction visible than points further away. Figure 6.14 shows the impact on accuracy resulting from using Weight instead of WeightF.

Although a linear fit of estimated (Weight) visibility fractions to benchmark



**Figure 6.14: N37E127 - Weight and R2 Visibility for 1024 Random Points.**

(R2) visibility fractions is maintained, the correlated variation has decreased from 0.98210 in Figure 6.11 to 0.74418; the y-axis intercept has not drifted greatly but the proportional term has decreased by about 10%, suggesting the Weight is over-estimating average visibility for this case much more than WeightF did. This is in accordance with our expectations. The overall poorer performance of Weight when compared to WeightF suggests that its modestly lower computational requirements would not generally justify its use.

### 6.5 WeightF to R2 Correlation as a Function of Ray Density

By varying the number of rays used in running the WeightF algorithm and comparing the resulting correlation to the corresponding R2 results, we can observe how the accuracy of the estimated visibility provided by WeightF increases with computational effort.

In Figure 6.15 we can observe a strong fitted relationship between the number of rays employed in producing WeightF visibility estimates and the correlated

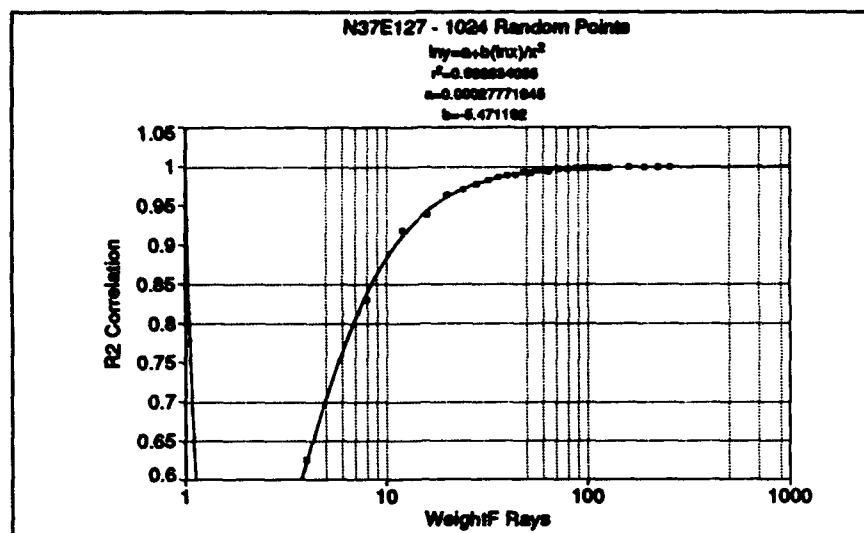


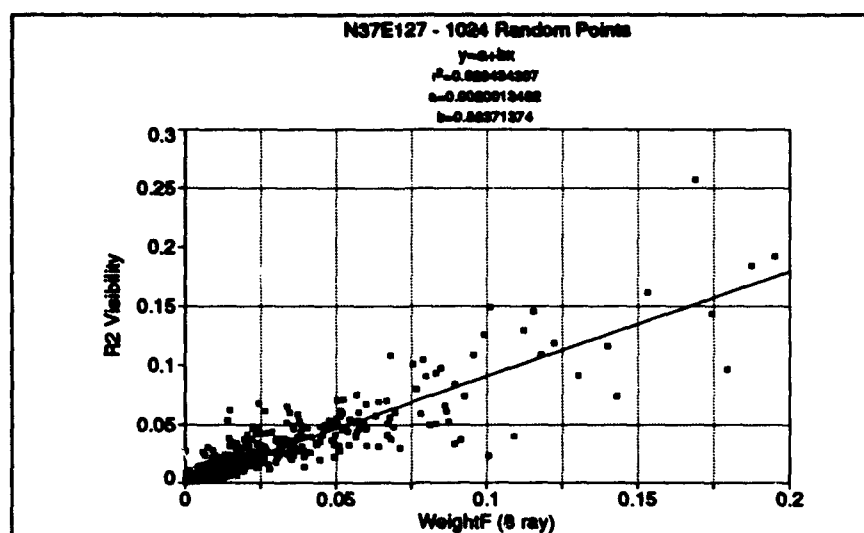
Figure 6.15: N37E127 - Correlation Variation of R2 to WeightF.

variation observed between the corresponding WeightF and R2 values (based on the  $y = a + bx$  model). The lowest number of rays employed in this sample was four, and the ascending 'tail' of the curve as the number of rays approaches 1 should be discounted as an artifact. The correlated variation of 0.99883 observed using the equation

$$\log y = a + (b \log x)/x^2 \quad (6.1)$$

is suggestive of both the logarithmic relationships often encountered in terrain-based characteristics and the inverse square relationship between many statistical sample sizes and the accuracy of the results derived from those samples.

As a specific instance of the manner in which the WeightF estimates degrade as ray density decreases, we can examine the plot of comparative WeightF and R2 results in Figure 6.16 using a ray density of 8. The overall correlation has decreased from 0.98210 in the 32 ray example from Figure 6.11 to 0.82943, with increasing magnitudes of error in the higher visibility regions. An examination of residuals shows the error in the WeightF estimates remains approximately proportional to



**Figure 6.16: N37E127 - Correlation Variation of R2 to WeightF (8 rays).**

visibility magnitude over the entire set of observed values. Referring back to Figure 6.15, we see that we could increase the correlated variation between WeightF and R2 values from 0.82943 to 0.91795 by increasing the ray count by 50% to 12 rays from each observer.

## 6.6 Analysis of Other Locales

### 6.6.1 Dijon Area

A natural question regarding the results presented so far is to what degree they are specific to the particular terrain (cell N37E127) that has been used for experimentation. It is a relatively complex region on the Korean Peninsula, proximate to the Yellow Sea, which has been shaped by geological forces much different from those in many other areas of real terrain. For comparison, an analysis is presented in Figure 6.18 that uses terrain from the DTED cell N47E005 in the vicinity of Dijon, France. Elevation ranges from 176 meters to 635 meters, with a median elevation

of 257 meters above mean sea level. The mean elevation is 280.4 meters, with a standard deviation of 76.8 meters.

The gray-scale elevation image of cell N47E005 is shown in Figure 6.17. This image was generated by median sampling 6 point (east-west) by 4 point (north-south) subcells in the elevation data for each image pixel. This compensates to within 5% for the unequal average east-west (92.61 meters) and north-south spacing (62.57 meters) between grid points.

This terrain encompasses plains regions from the Saône and Doubs rivers, as well as higher plateau and mountain areas. The same  $(x, y)$  pseudorandom coordinates used for the examples from N37E127 were used here to minimize the impact of differences in the spatial distribution of sample points as a source of variation.

The general appearance of the sample data, the  $y = a + bx$  form of the fitted equation, and the correlated variation of 0.9922 all compare favorably and consistently with the values from the data in Figure 6.11 for the N37E127 cell. The constant and proportion terms also suggest a very consistent relationship between WeightF and R2, without substantial over- or underestimation. The improvement in the 32 ray WeightF to R2 correlated variation from 0.9821 for the N37E127 terrain to 0.9922 conforms to our subjective expectations, given the less complex nature of the terrain in the Dijon area.

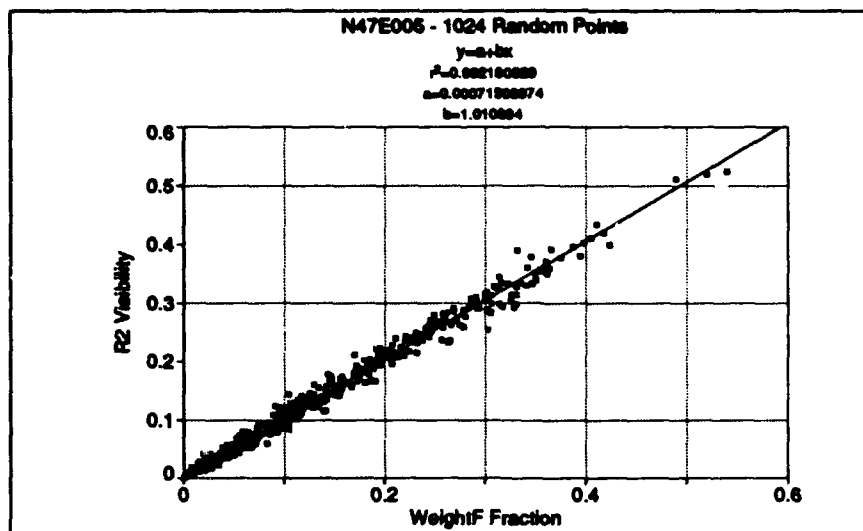
### 6.6.2 Pripyat' Area

As a final local to investigate the performance of the WeightF algorithm and the properties of the resulting visibility output, an analysis was conducted for the one degree DTED cell N52E028 (also drawing on its eight surrounding cells as required), which is near the center of the Pripyat' Marshes in Beloruss, northwest of the Ukrainian city of Kiev. Figure 6.19 shows the elevations of cell N52E028 as a gray-scale image. This cell was median-sampled on 4 point by 5 point subcells.



**Figure 6.17: N47E005 - Gray-Scale Elevation Image.**





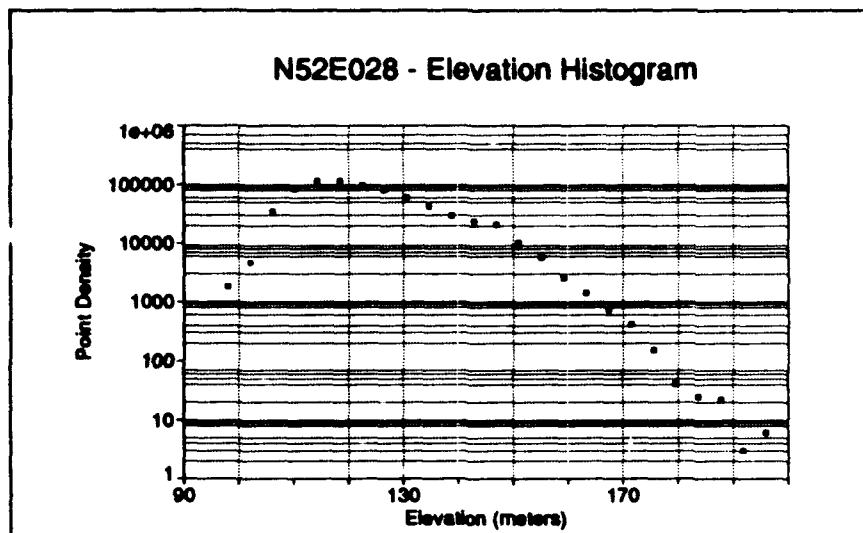
**Figure 6.18: N47E005 - WeightF and R2 Visibility for 1024 Random Points.**

The N52E028 cell differs significantly from previous examples in two ways. First, the longitudinal spacing changes from 3 arc seconds to 6 arc seconds, resulting in half as many longitudinal divisions in the cell with an average longitudinal distance between points of 112.76 meters instead of distances in the 60–75 meter range as in previous examples. Due to the decreased longitudinal density, each cell will have only 721,801 observer locations instead of the 1,442,401 locations for cells in previous examples. In addition, the N52E028 area consists of terrain with elevations that are relatively low and uniform. Elevations only range from 96 to 198 meters above mean sea level, with a mean elevation of 123 meters and a standard deviation of only 11.5 meters. Median elevation is 121 meters. In spite of the reduced standard deviation and range of elevations when compared to N37E127, the elevation histogram of N52E028 in Figure 6.20 shows the same log-linear form over most of the range of elevations as was shown in Figure 3.2.

N52E028 was selected to see how the accuracy of the WeightF visibility estimates and the distribution of visibility values would be affected by relatively low,



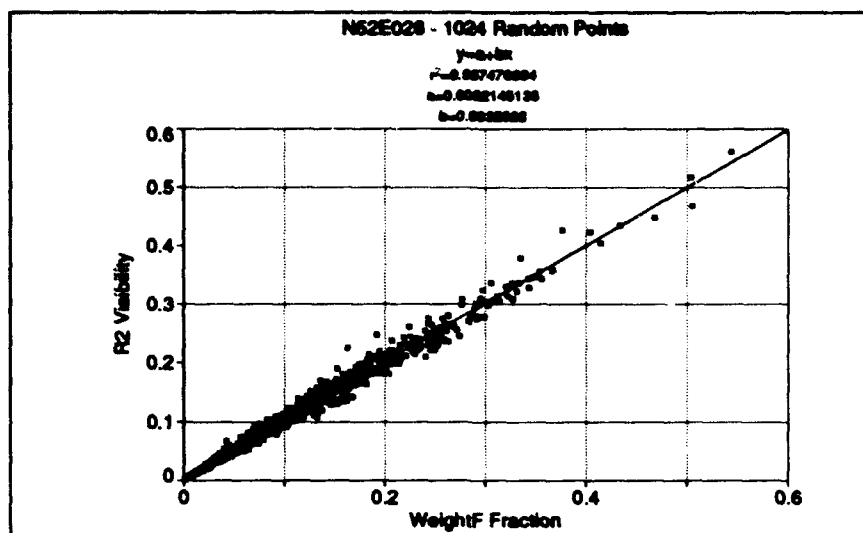
**Figure 6.19: N52E028 - Gray-Scale Elevation Image.**



**Figure 6.20: N52E028 - Elevation Histogram for 1024 Random Points.**

uniform terrain. Intuitively, we might suspect that the accuracy of the estimates and overall visibility would increase due to the less complex nature of the terrain. Figure 6.21 shows the results of running the WeightF algorithm on the same pseudorandom spread of 1024 points used in previous examples.

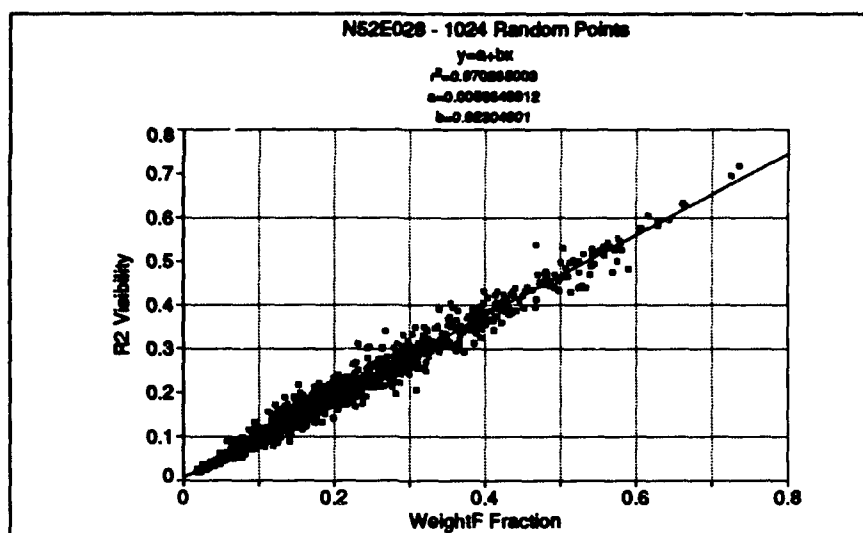
The same strong linear trend that has been observed in previous examples is still present. As expected, the range of visibility fractions extends into significantly higher values. For the complex terrain in cell N37E127, in Figure 6.11 the highest observed R2 visibility fraction is 0.25646 and for N52E028 (as shown in Figure 6.21) this has increased to 0.56028, more than doubling. In spite of the general increase in visibility, however, very high visibility points (for example, those above an R2 visibility fraction of 0.4) still occur in very small numbers (also shown in Figure 6.23). The correlated variation of 0.987477 is somewhat higher than the corresponding value of 0.982097 obtained in cell N37E127, which is in accord with the hypothesis of the impacts of the greater spatial cohesion present presumed present in N52E028.



**Figure 6.21: N62E028 - WeightF and R2 Visibility for 1024 Random Points.**

To investigate how visibility relationships change with differing LOS parameters, Figure 6.22 displays the results obtained by performing the same calculations on the same points as in Figure 6.21 but with an alternative set of LOS parameters. This parameter set uses a radius of visibility of 4 *km* instead of 40 *km*, an observer height above ground of 1 meter instead of 5 meters, and a target height above ground of 3 meters instead of 25 meters. This set of parameters is more representative of visibility problems involving human vision instead of antennas.

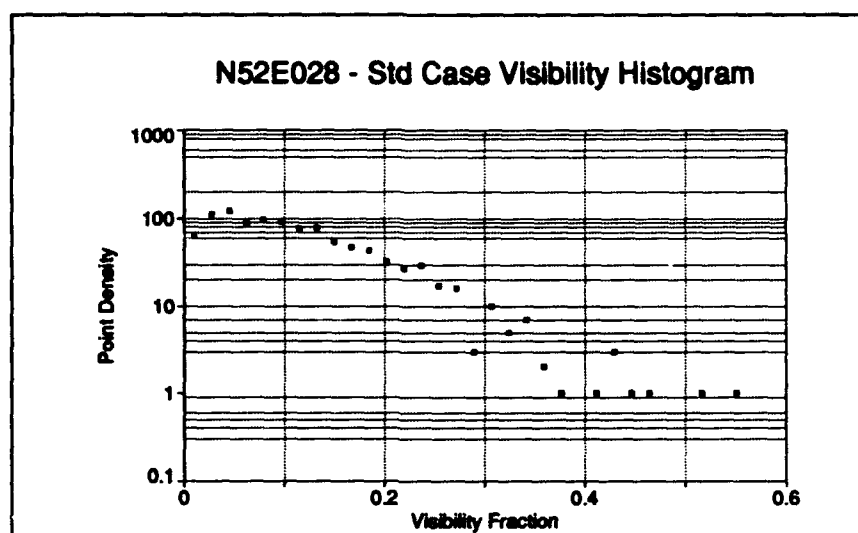
Using this new set of parameters, a strong linear relationship between the WeightF estimates and the R2 benchmark values remains. The correlated variation has decreased slightly from 0.987477 using the original parameters to 0.970265 using the alternative parameter set. A greater change is observed in the constant of proportionality. This decreases from 0.99329 to 0.92305 and suggests that under these conditions of a smaller potential viewshed, WeightF is still very successful in assigning relative visibility estimates but overestimates visibility as measured by R2. Still apparent is the pattern of very small numbers of points with high visibility



**Figure 6.22: N52E028 - WeightF and R2 with Alternate LOS Parameters for 1024 Random Points.**

values (for example, R2 values above 0.6). We also note a modest increase in the maximum visibility value from 0.56028 using the original parameters to 0.71719 under the alternative parameter set.

Figure 6.23 shows a histogram of R2 visibility values from N52E028 using the original LOS parameter set. All visibility observations were categorized into 32 equally spaced slots. Use of a log scale on the y-axis brings out a strong linear trend over much of the range of plotted points, corresponding to a decreasing exponential relationship between visibility values and the number of points. Log-linear regions are common in the distributions of many real world characteristics ([FR94a]) and reflect a product-form instead of an additive relationship between underlying contributing factors. For example, two factors contributing to the total visibility from an observer location are the configuration of the terrain in its immediate vicinity and the extent of visibility barriers at intermediate ranges. Neither factor by itself is adequate to ensure high visibility at longer ranges where the majority of the total viewshed lies; only when both of these factors are favorable can a high total visibility

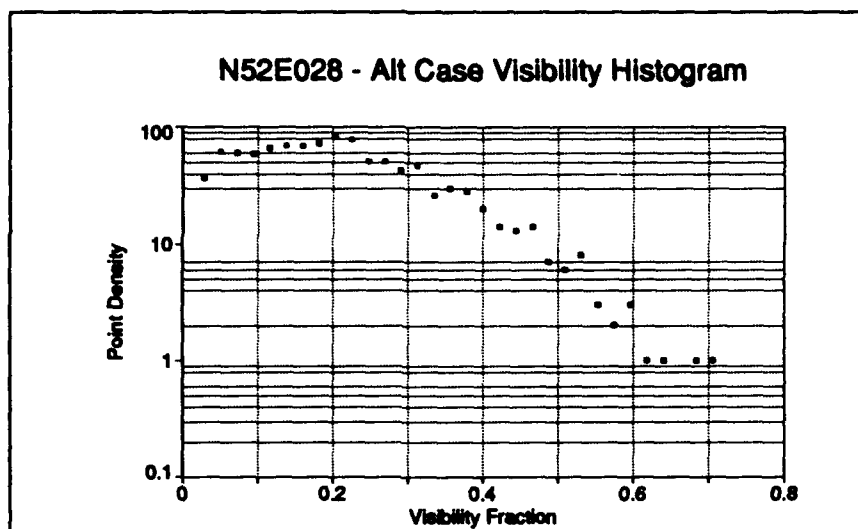


**Figure 6.23: N52E028 - Histogram of R2 Visibility for 1024 Random Points**

value occur. The impact of both factors being favorable has a much greater than additive effect in producing a high count of total points visible.

The data displayed in Figure 6.23 support the subjective observations from previous figures that a relatively small set of points will have visibility values distinguishably higher than the bulk of points in a given area. This relationship reinforces the value of searching through an entire area for the points with highest visibility when siting small numbers of observers.

We can see the trend between visibility and frequency of points again in Figure 6.24, which is a histogram similar to that in Figure 6.23 but for the alternative set of LOS parameters. This figure also displays substantial linearity in the visibility range between 0.2 and 0.6 when plotted using a y-axis log scale. The onset of this trend at a value of 0.2 occurs at a higher visibility than in Figure 6.23, where it occurred at approximately 0.05. This is consistent with the generally higher visibility values present in the alternative LOS parameter setting.



**Figure 6.24: N52E028 - Histogram of R2 Visibility with Alternate LOS Parameters for 1024 Random Points**

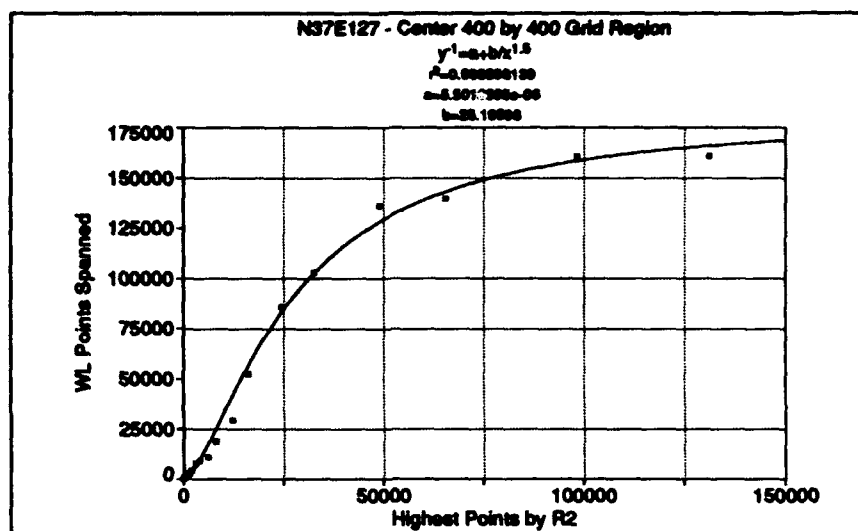
### 6.6.3 Summary

The WeightF algorithm is a highly effective and efficient modification to the basic VL algorithm, extending the predictive power of VL for high visibility points to the entire range of points.

## 6.7 Confidence in Finding Highest Points

To further verify the accuracy of the WeightF algorithm and to investigate how many of the points with the highest WeightF visibility index estimates should be selected in order to have high confidence that the  $n$  highest 'true' (R2 valued) visibility points had been included, an exhaustive study of the central 400 by 400 grid region of the DTED cell N37E127 was undertaken. This required approximately 45 days of CPU time to calculate R2 values for each point; actual execution was shared across several workstations. Execution time for calculating WeightF values was approximately 6.1 hours.

One practical measure of the utility of the WeightF algorithm in finding the



**Figure 6.25: Center of N37E127 - WeightF Points Required to Find the Highest R2 Points**

$n$  highest visibility points in a region is given by determining how many of the top WeightF points in addition to  $n$  must be evaluated by R2 to have high confidence that all of the true highest  $n$  valued points have been found. Figure 6.25 shows how many WeightF points had to be evaluated to find a given number of top R2 visibility points in the central 400 by 400 grid of N37E127. From the inflection of the curve it is evident that the ratio of WeightF points required to R2 highest points desired is highest in the region located approximately between 10000 and 100000 highest points. This can be seen more clearly by plotting the ratios of required to desired points, which is done in Figure 6.27. It is useful to bear in mind when examining the relationship between desired and required points that in practical siting problems, the number of locations to be selected for analysis will be very much less than the total number of points in the elevation grid.

Figure 6.26 suggests that the number of highest visibility points that have values in the top 50% of the total range of visibility values encountered may be only a few dozen out of over a 160,000 total points. In fact, in the top third of the total



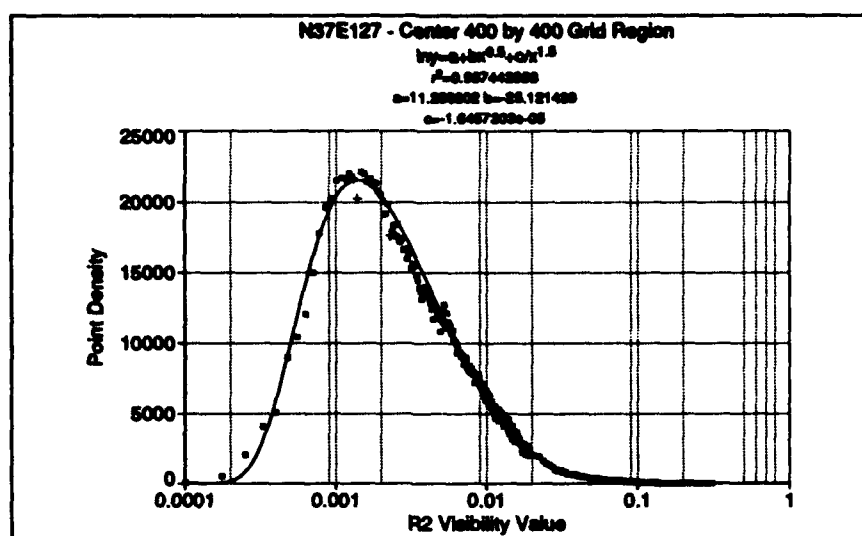


Figure 6.26: Center of N37E127 - Histogram of R2 Visibility Values

range of visibility values, only 50 points occur.

While the general shape of the observations in Figure 6.26 suggests a log-normal distribution, if such a distribution is fitted to the peak in the data it is too high for values to the left of the peak and too low for values above it. Nevertheless, the fitted curve of  $\log y = a + b\sqrt{x} + c/x^{\frac{1}{2}}$  does exhibit many of the characteristics of the log-normal distribution.

Figure 6.27 shows the variation between the required number of highest WeightF points to find a given number of highest R2 points by plotting the ratio of the two versus the number of highest R2 points desired. Note that the ratio of WeightF points required to find a given number of highest R2 points never exceeds 4, and for the range of R2 highest points from 1 to 10000 it never exceeds 3. Using more than 32 rays in the WeightF estimated visibility values would be expected to reduce the ratios of required WeightF to desired R2 highest points, and a tradeoff in execution time between number of WeightF rays and subsequent R2 evaluations could be performed on specific hardware and problem settings.

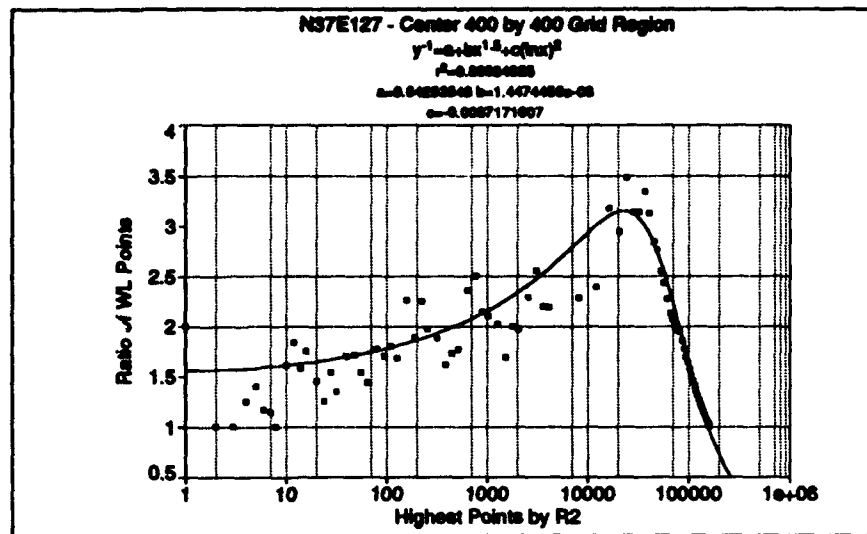
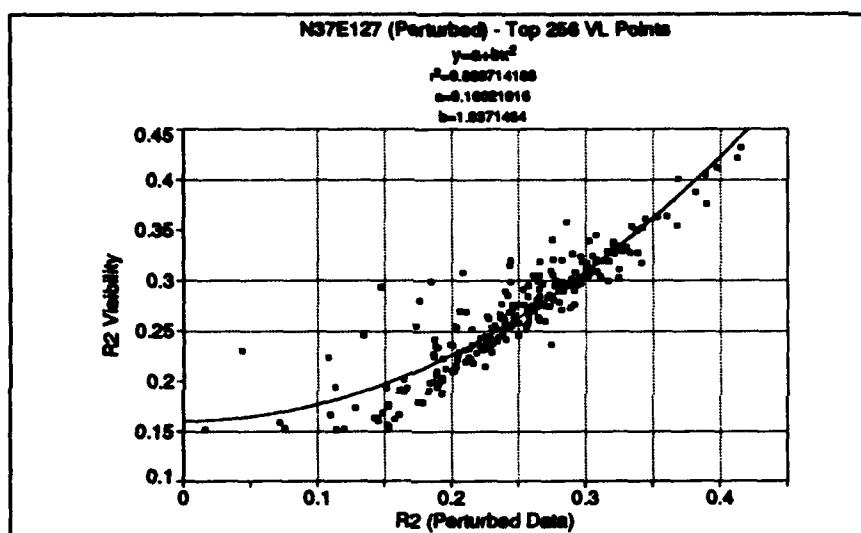


Figure 6.27: Center of N37E127 - Ratio of WeightF Points Required to Number of Top R2 Points

### 6.8 Impact of Errors in Elevation Data

One reason why evaluating the impact of errors in elevation data is important is because it can help evaluate the relative significance of other uncertainties in the results produced by particular visibility estimation techniques. In [Fis92] one conclusion was that introducing simulated error reduces viewsheds. This is consistent with the concept that the existence of substantial visibility is related to the presence of significant spatial correlation in real terrain. As an experiment, elevations for the cell N37E127 and its surrounding cells were perturbed pseudorandomly from a Gaussian distribution with mean of zero, standard deviation of 7 meters, and with no introduction of spatial correlation in the errors.

Figure 6.28 shows clearly that the points least impacted by the introduction of errors in the elevation data are those points with the highest visibility values. This lends credence to the use of consistent and unbiased visibility estimation techniques for the purpose of identifying the points in an area with the highest visibility values



**Figure 6.28: N37E127 - R2 Visibility for Original and Perturbed Elevations**

in spite of the presence of errors in the elevation data.

The existence of efficient algorithms to calculate visibility fractions (R2) and to identify points in with the highest visibility values (WeightF) makes it possible to make inferences such as mentioned above regarding Figure 6.28, because large numbers of points and their visibility results can be examined, calculated, and aggregated.

## 6.9 Summary

In this chapter, we have evaluated the accuracy of several algorithms that use a fixed number of LOS rays to estimate the visible fraction of a potential viewshed from a given observer. The R2 algorithm was used to provide the reference value for viewshed fraction. It was determined that the most direct LOS ray algorithm, VL, was useful in identifying points with high or low relative visibility, but had a correlated variation of only 0.776 using the standard data cell and visibility parameters. It suffers from a non-linear systematic overestimation of visibility that is

most severe for points of moderate visibility. An algorithm which weights each point along an LOS ray by its distance from the observer produces excellent viewshed fraction estimates, with a correlated variation using the standard data cell and visibility parameters in excess of 0.98 and relatively uniform performance for the entire range of visibility values. This performance was evaluated using a variety of LOS ray densities, three different geographic settings, and an alternative set of visibility parameters.

## CHAPTER 7

### EXECUTION TIMES

#### 7.1 Overview

In this chapter we investigate the execution times of the R3, R2, and LOS ray estimation algorithms and compare empirical measurements to predicted performance

A sample set of 256 points was drawn from the standard set of 1024 pseudorandomly generated coordinates described earlier. These points were sorted into ascending order according to data layout to minimize the impact of paging on the timing results. Timing values were based on the sum of 'user' time and 'system' time reported by the UNIX 'time' command. The output of this command is intended to measure time spent by processors in running or supporting (e.g., paging) the execution of the problem program, regardless of 'real' (wall clock) time. Each run was executed as one process on a 4-processor Sun 670-MP. Total available real memory was 128MB.

To reduce variations in timing measurements unrelated to running the visibility calculations, several precautions were taken. Runs took place on identically configured Sun 670-MP computers with negligible workloads (network daemons, etc.). Output volume was minimal — a few hundred characters of textual summary information. A delay of 180 seconds was imposed between successive runs to reduce the impacts of coincidental cached resources from one run to the next. To minimize the effects of possible caching of disk-based input data by the operating system, the 25.9MB input file was kept on the disk system of a separate Sun 670-MP and accessed by means of remote mounting using the Network File System (NFS) protocol.

## 7.2 R3 Execution Time Performance Evaluation

We should expect that using the R3 algorithm we will observe startup time (loading the elevation file, adjusting the effective heights of the elevation data for curvature of the earth, establishing a working set in real memory, etc.) to be relatively constant between all runs, and the time of 225.2 seconds for the run with the shortest radius (1250 meters) provides an upper limit for the estimated value of the startup time. The execution time for the R3 algorithm should vary with the cube

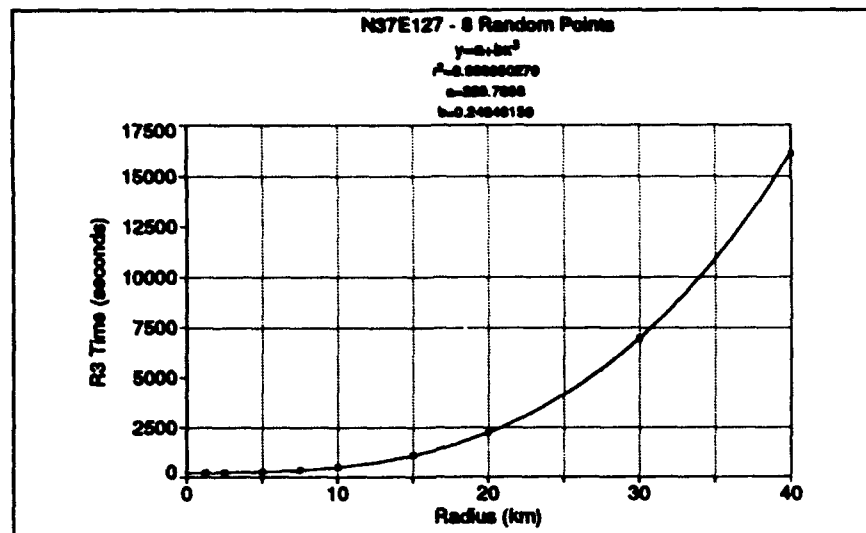
Radius (km)	Run Time (seconds)	Fitted Model Time(seconds)
1.25	225.2	230.3
2.50	229.4	233.7
5.00	259.0	260.8
7.50	335.4	334.6
10.00	487.3	478.3
15.00	1088.2	1068.4
20.00	2243.3	2217.6
30.00	6944.1	6938.8
40.00	16084.4	16132.6

**Table 7.1: R3 Execution Times as a function of the Visibility Radius**

of the radius. Figure 7.1 shows the run times for the R3 algorithm using conditions similar to those used to compile execution time data on R2, with the exception that 8 points were used instead of 256 on each run and longer ranges were not measured due to the excessive run times required.

To account for knowledge that the run times of the shorter radii form a ceiling on the startup time, the  $y = a + bx^3$  form of the fitted equation was selected by using raw timing data. Then, in order to more properly reflect the significance of the lower radii run time values in establishing a ceiling on startup time (the value of the  $a$  parameter), the fitting procedure was run again with each point weighted by the reciprocal of its run time.

The fit to weighted data is what is displayed in Figure 7.1. Weighting the



**Figure 7.1: N37E127 - 8 Random Points; Weighted Curve Fit**

fit changes  $b$  from 0.24776 to 0.24848 and decreases the correlated variation from 0.99999296 to 0.9999503, but brings the value of  $a$  (which should reflect average startup time) from 238.3 to 229.8 seconds, with 95% confidence interval boundaries of 237.0 and 222.6 seconds. This is a noticeable improvement in agreement with the two lowest run time values of 225.2 and 229.4 seconds. The high correlated variation using either weighted or unweighted data for a fitted curve of the form  $y = a + bx^3$  confirms the anticipated cubic relationship between the run time of the R3 algorithm and the radius of visibility.

### 7.3 R2 Execution Time Performance Evaluation

The number of points within the visibility radius for which the R2 algorithm must make a visibility determination is shown in Figure 7.2 as it varies with radius length. Use of log scale on both axes reveals a relatively straight line relationship that brings out the simple power relationship (in this case quadratic) between radius and points within the boundary followed by R2.

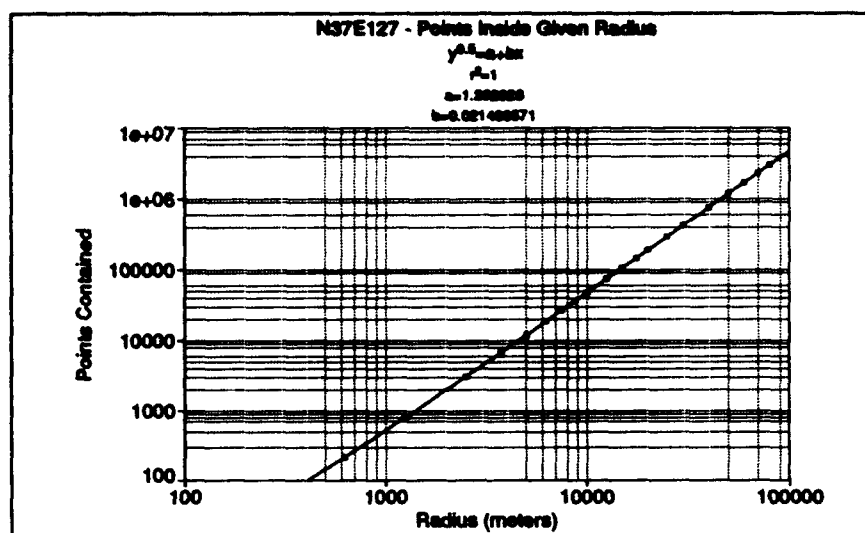


Figure 7.2: N37E127 - Points within R2 Radius

Note that the radius units are meters, not grid points. For the cell N37E127 the average spacing in meters between points is 73.475 meters per 3 arc seconds of longitude and 92.614 meters per 3 arc seconds of latitude. The relation between the radius and the points within it takes the anticipated form, in that

$$\sqrt{y} = a + bx \quad (7.1)$$

is tantamount over our range of interest to

$$y = a' + b'x + cx^2 \quad (7.2)$$

where

$$a' = a^2 = 1.5942$$

$$b' = 2ab = 0.054259$$

$$c = b^2 = 0.00046167$$

The non-zero y-intercept of 1.5942 corresponds to an intuitive limiting case of one point for a zero radius, that point being the observer location itself. The



number of LOS rays computed by R2 is determined by the number of boundary points (see Chapter 4), which is in turn proportional to the circumference around the area of the points within the given radius of visibility. Since the number of operations involved in computing a single LOS is also proportional to the visibility radius, the work required in the R2 algorithm in running all LOS's to the boundary points is proportional to the square of the radius. As shown above, the number of operations to set visibility values for interior points (based on results from running LOS's to the boundary) is also quadratic in the visibility radius. Therefore, in terms of performance we expect that the execution time of R2 would vary with the square of the visibility radius, after accounting for program startup time. In Figure 7.3 and Table 7.2 we can examine this hypothesis in the context of actual run data.

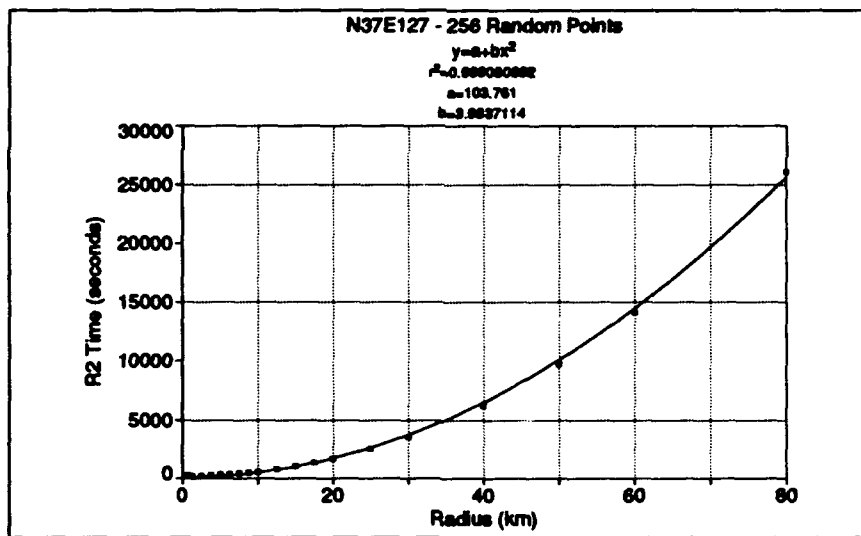


Figure 7.3: N37E127 - R2 Execution Times for 256 Random Points

Superficially, Figure 7.3 suggests that a good fit between the observed data and the predicted quadratic behavior of the R2 algorithm has been achieved, fitting the expected form of

$$y = a + bx^2 \quad (7.3)$$

The correlated variation of 0.99908 seems high, but there are warning signs that an improved fit can be found. For radius values in the range between 17.5km and 60km the fitted curve consistently overestimates the observed data. For radius values less than 17.5km, the fitted curve consistently underestimates the observed data. In particular, the y-intercept value of 103.8 seconds substantially underestimates the clear evidence of the points with the smallest radius values that the minimum execution time (due to startup operations) is very likely to lie between 200 and 250 seconds.

Radius (km)	Run Time (seconds)	Fitted Model Time(seconds)
1.25	227.3	223.5
2.50	242.3	240.1
3.75	268.4	267.8
5.00	302.8	306.6
6.25	356.1	356.6
7.50	411.6	417.6
8.75	485.6	489.7
10.00	563.4	572.9
12.50	775.4	772.5
15.00	1012.1	1016.5
17.50	1305.2	1304.9
20.00	1643.0	1637.7
20.00	1643.0	1688.7
25.00	2491.2	2479.0
30.00	3487.6	3477.6
40.00	6187.4	6124.6
50.00	9682.1	9672.6
60.00	14087.8	14155.8
80.00	26058.9	26039.8

**Table 7.2: R2 Execution Times as a function of the Visibility Radius;  
Separate Fitted Curves for each Group**

By experimenting with breaking the observations into two groups, varying the break point, and evaluating the curves fitted to each group, substantially better results were obtained. The results of the separate fits are presented in Table 7.2

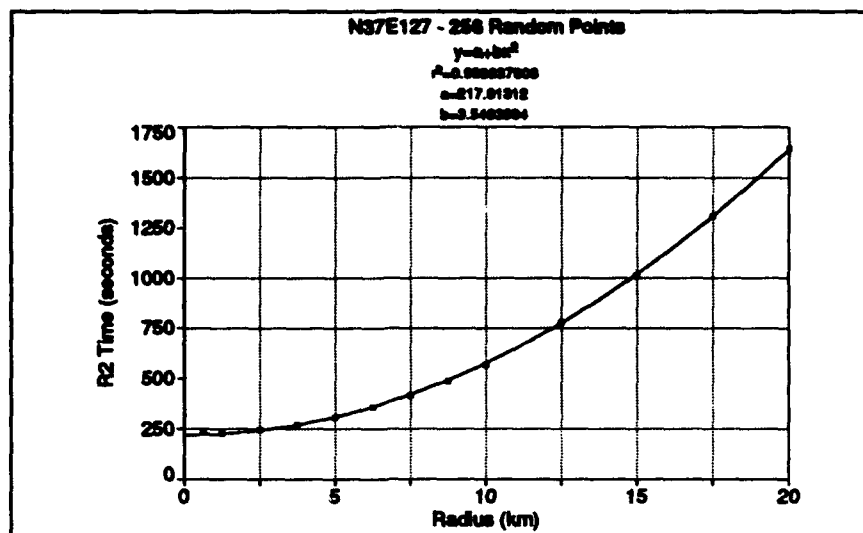
and in Figures 7.4 and Figure 7.6. A breakpoint of 20km was selected by examining how the F-statistic for each of the fitted curves varied as the breakpoint varied.

For points with radius values in the range between 0 and 20km, the form of the fitted equation remains  $y = a + bx^2$ . The y-intercept value has changed from 103.8 to 217.9, which is a significant improvement in the estimate of the minimum execution time over the previous value obtained by fitting a curve to the entire range of points. The correlated variation has improved from 0.99908 to 0.99984, representing a nearly five-fold decrease in the amount of uncorrelated variation remaining.

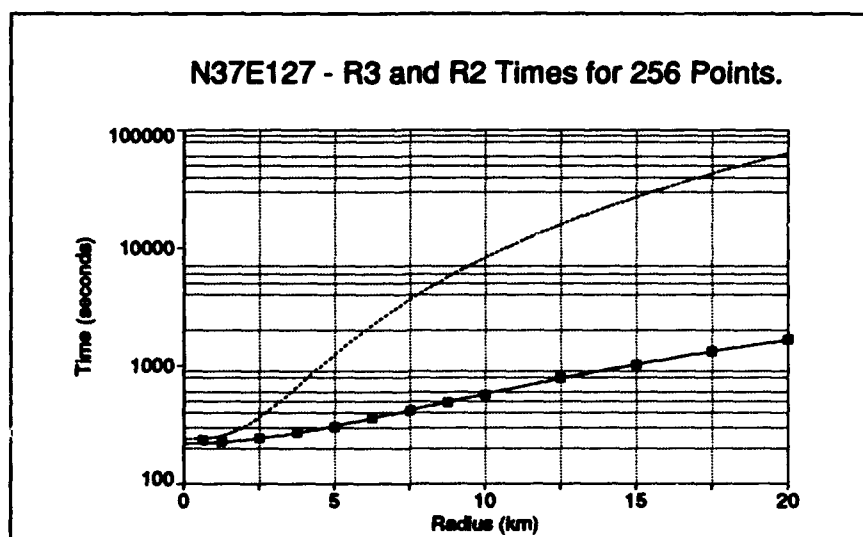
An examination of the residuals between the observed and modeled execution times also shows improvement when specifically fitting the 0 to 20km range. When fitting the entire range, the trend of the residual percentages forms a smooth, continuous pattern of decreasing values from 0 to 30km, with the first 10 of the 12 total observations between 0 and 20km being positive. After fitting  $y = a + bx^2$  to the range of 0 to 20km, the residuals pattern becomes much more erratic, a sign of an improved fit. The longest run of residuals with the same sign is now 5 instead of 10, with an even total split of 6 positive and 6 negative residuals.

The constant of proportionality on the  $x^2$  term on the equation fitted in Figure 7.4 has decreased to 3.55 from the value of 3.98 obtained when fitting the entire range of radius values. This is consistent with the increase in the y-intercept value while improving the fit with all of the data points in the 0 to 20 km range.

For convenience, Figure 7.5 shows the execution times for R3 and R2 from 0 to 20 km on the same graph, using a logarithmic scale for time. The dotted line represents the time performance for R3 for 256 points, derived from the curve fitted to R3 observations for 8 points in Figure 7.1 by multiplying the constant on the  $x^3$  term by 32. The solid line with boxes indicating observed values represents the same R2 information depicted in Figure 7.4. The pronounced time advantage of R2 over R3, especially for longer radius values, is readily apparent.



**Figure 7.4: N37E127 - R2 Execution Times for 256 Random Points (lower range)**



**Figure 7.5: Comparison of R3 and R2 Execution Time for 256 Points.**

Figure 7.6 and the second portion of Table 7.2 show a significant improvement resulting from separately fitting the range of points between 20 and 80km. When fitting the entire range of points, the 7 residuals in the 20 to 80km range were negative for the first 6 observations. By specifically fitting the 20 to 80km range the residuals split to 5 positive and 2 negative values, with four positive values being the longest consecutive constant sign run of values. The correlated variation has increased from 0.99908 to 0.99997, representing a 30-fold decrease in uncorrelated variation.

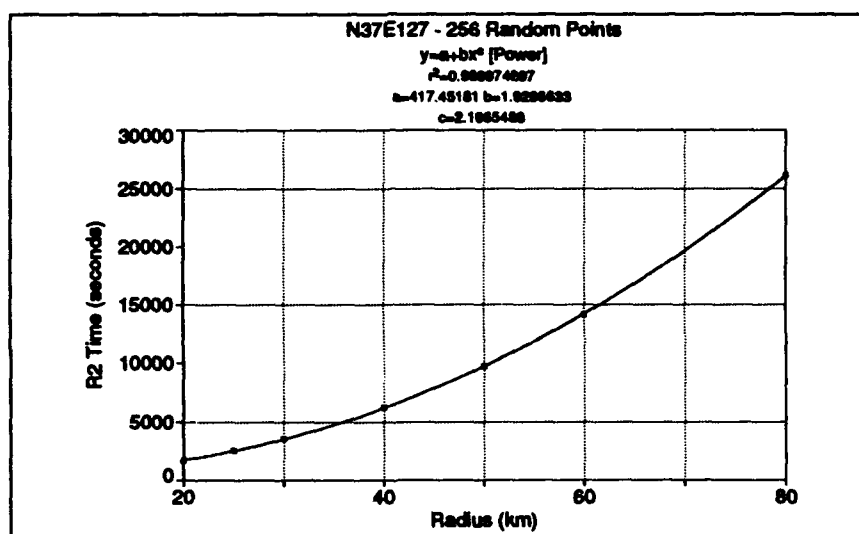


Figure 7.6: N37E127 - R2 Execution Times for 256 Random Points (upper range)

The most significant item to note about the curve best fitted to the data values in the 20 to 80km range is that the form of the fitted equation has changed from

$$y = a + bx^2 \quad (7.4)$$

to

$$y = a + bx^c \quad (7.5)$$

Although the power coefficient value  $c = 2.1665$  is not far from the value of  $c = 2$  for a simple quadratic relation, it represents a distinguishable difference from the original fit to the entire 0 to 80km range and to the improved quadratic fit on the 0 to 20km range.

Since the R2 algorithm performs the same pattern of steps regardless of the visibility radius, one other place to look for an explanation of the small upward deviation in run time is to the architecture of the underlying hardware. As with many current system architectures, in addition to system main memory the processor modules on a Sun 670-MP contain small on-chip data and instruction caches. There is also an external cache between each processor and system main memory. Although the on-chip processor caches are too small to account for any performance change observed for radii above about 2-5 km (depending on assumptions about cache operation), interactions between the problem size and the external cache may have some impact on execution time for larger radius values. Given a finite cache size and the known characteristics of the R2 algorithm, it is likely that Equation 7.5 represents a transition zone between shorter ranges where cache hits dominate data accesses and longer ranges where cache misses dominate data accesses. In each of these zones a quadratic relationship in the form of Equation 7.4 would describe the relationship between visibility radius and execution time, but with a higher  $b$  value at longer ranges.

An approximation of the possible relationship between external cache size and the radius value at which the problem size and access patterns may start to generate a substantially increased cache-miss rate (and therefore longer execution times) can be formulated as follows. We assume that most but not all of the external cache can be used to hold elevation data to speed processing of the visibility calculations, because some of the cache will be used to hold program code, operating system functions, or will be malutilized in the visibility calculations due to address mapping

or related conflicts. Since we assume that more than half but less than all of the cache will be used for the visibility calculations, we arbitrarily chose an estimate of 75% cache use for visibility calculation elevation data. Each elevation value requires two bytes to represent. The minimum number of points per kilometer in cell N37E127 is

$$10.8 \text{ points per km} = (1000 \text{ meters}) / (92.6 \text{ meters per North-South grid})$$

and the maximum number of points per kilometer is

$$13.6 \text{ points per km} = (1000 \text{ meters}) / (73.5 \text{ meters per East-West grid})$$

so we will use an average of 12 points per kilometer in this approximate analysis. Since we have an empirical indication of a change in run time performance characteristics at a radius of around 20km, this would correspond to an external processor cache size of approximately:

$$\begin{aligned} \text{Cache Size (bytes)} &= \pi r^2 \text{ points} \times 2 \text{ bytes per point} / \text{Cache Utilization} \\ &= \pi \times [20 \text{ kilometers} \times 12 \text{ points per kilometer}]^2 \times 2 / 0.75 \\ &= 482549 \text{ bytes} \end{aligned}$$

or approximately 471KB. This agrees with the 1024KB actual size of the external cache on the Sun 670-MP processors to within about a factor of two. This should not be taken as confirmation of the hypothesis regarding the size and manner in which cache size will affect the execution time of the R2 algorithm, but only as support for its plausibility. Further research using larger problem sizes and processors with easily controllable cache operation (such as 80486 or Pentium class personal computers) would be able to better determine the interactions between problem and cache size.

#### 7.4 WeightF Execution Time Performance Evaluation

The WeightF algorithm exhibits a strong linearity in execution time for visibility ranges out to 60 km. The same measurement methodology used for measuring

the performance of R2 and R3 algorithms was employed, with the exception that the 256 point sample set was executed 16 times during each run to obtain longer times that stood out better from 'noise' variations for shorter ranges. Figure 7.7 shows the accurate linear fit.

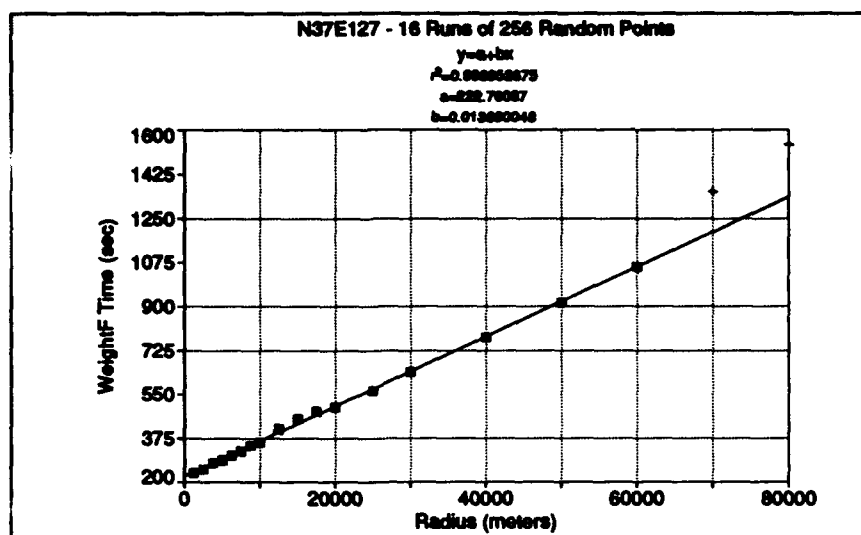
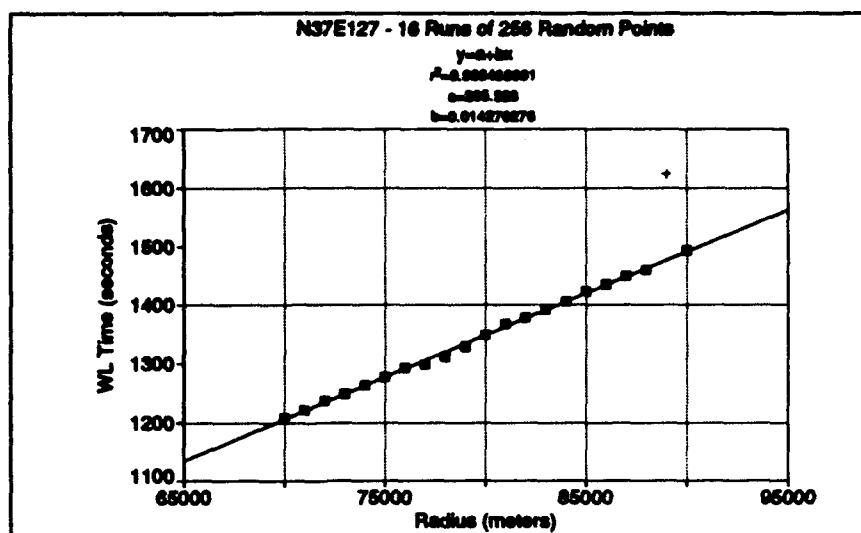


Figure 7.7: N37E127 - WeightF Execution Times for 16 runs of 256 Pseudo-Random Points (out to 60 kilometers)

The same break from the linear trend at longer ranges occurred for WeightF as was observed for R2, although at a longer specific range (60 *km* instead of 20 *km*). This break is clear when comparing the 70 *km* and 80 *km* values (which were excluded from the linear fit) to the fitted line.

A different phenomenon here is responsible for the higher than expected execution times at higher ranges than was observed for execution time for R2 at longer ranges. The processor time devoted to system functions (approximately 140–170 seconds) is much higher for the 70 *km* and 80 *km* values in Figure 7.7 than for runs at smaller ranges (approximately 25–35 seconds), which was not the case for R2 runs. This is more likely to be attributable to paging operations than memory cache behaviors.





**Figure 7.8: N37E127 - WeightF Execution Times for 16 runs of 256 Pseudo-Random Points (from 70 to 90 kilometers)**

Figure 7.8 shows observations of WeightF execution times conducted for visibility ranges between 70 *km* and 90 *km*, with a 1 *km* change in range between adjacent runs. These runs were conducted in parallel on separate processors on the same Sun 670-MP in groups of 10 or 11 consecutive ranges per processor, with each run being preceded by a 'dummy' run at the lowest range in the group in an attempt to establish a common base of buffered data files, etc.

Starting out each sequence of runs with a 'dummy' run and using a fine range granularity (1 *km*) produces a sequence of execution times for the 70 *km* to 90 *km* range much more in accord with the observations for lesser ranges from Figure 7.7, with the exception of the observation for the 89 *km* range (this point was manually excluded from the curve fit). The system time for this point was approximately 140 seconds, more than 100 seconds above the system times observed for the remainder of the observations. The user time for this point is also somewhat higher than what would be predicted by the fitted curve. If the jump in system time is caused by increased paging requirements to bring the elevation data required by the longer

LOS lengths into memory, this modest jump in user time may be due to increased context switching charged to the user process, increased cache misses, or other inefficiencies.

### 7.5 Summary

Strong empirical evidence exists that the predicted execution times for R3 [ $\Theta(R^3)$ ], R2 [ $\Theta(R^2)$ ], and WeightF [ $\Theta(R)$ ] match the behavior observed running the algorithms using real data on physical hardware. Anomalies have been observed for some points or transition zones which may be due to caching or paging behavior.

## CHAPTER 8

### SUMMARY AND CONCLUSIONS.

The potentially immense demands involved in computing visibility and the utility of visibility analyses makes finding ways of reducing the computational complexity of suitable visibility algorithms an important problem. We have focused here on developing and validating algorithms that are accurate, significantly faster than existing methods, practical to implement, and robust on real world data. This chapter summarizes methods and results, and then lists some areas for extensions and future research.

#### 8.1 Summary

To facilitate development and evaluation of algorithms dealing with terrain datasets of significant size, a variety of access routines coded in C++, an object-oriented third generation computer language, were written, tested, and refined. These routines provide transparent access to five types of USGS data, DMA (Defense Mapping Agency) DTED Level I data, TEC (U.S. Army Topographic Engineering Center) TerraBase data, and to any other raster based layout that can be described by parameters such as grid orientation, geographic location, ground spacing, etc. A uniform format for representing datasets of gridded elevation and visibility values was devised to provide for efficient storage and access.

To be able to investigate and validate very fast  $[O(R)]$  LOS ray methods of computing visibility indexes, it was necessary to develop an accurate viewshed computation algorithm that was substantially more efficient than  $[O(R^3)]$  naive methods such as those used in the common R3 algorithm. The R2 algorithm  $[O(R^2)]$  developed here meets these requirements. For the standard case of a 40 km viewshed

in the DTED Level I cell N37E127 with common startup time discounted, viewshed calculation time on a Sun 670-MP for 256 observer locations using R3 takes approximately 5.89 days but only 1.66 hours using R2.

In order to produce accurate visibility index values in  $\Theta(R)$  time, a method to exploit the spatial correlation of real terrain was designed. This used a fan of equally spaced LOS rays from observer points to sample the visibility of the surrounding terrain and produce an estimated visibility index.

The initial LOS ray algorithm, VL, was evaluated on a large number of terrain datasets and found to be useful in identifying points of high visibility but was less accurate for other points and tended to overestimate true visibility index values (as measured by R2). Investigation and experimentation demonstrates that a weighted version of the VL algorithm, WeightF, provides very accurate (correlated variation to R2 above 0.9) visibility index estimates over the entire range of values for only a minimal constant time increase in computation. Accurate estimates for visibility index values requiring approximately 1.66 hours to compute using R2 (see above) can be calculated in less than 10 minutes using WeightF.

To gain confidence in the visibility algorithms developed, experiments were run on terrain from several parts of the world ranging from mountainous to river valley to marshland. Over two dozen DTED Level I cells were analyzed, each with over 1.4 million observer locations and viewsheds of over 740,000 points from each observer. Parallelism in various degrees was employed to spread the work over several workstations and processors.

Verification of predicted execution time behavior as a function of problem parameters was performed by controlled runs of the algorithms on several platforms. Some variation from expected timing results for R2 was observed, which may be attributable to memory cache characteristics.

## 8.2 Significant Contributions and Accomplishments

There are several contributions realized by the results of the research described in this thesis.

- The existence of significantly faster algorithms for viewshed and visibility index calculation broadens the range of applications that can be developed and the scope and nature of further visibility research that can be undertaken.
- The existence of efficient and implementable viewshed and visibility index algorithms provides benchmarks for developing even faster methods based on genetic algorithms, hierarchical data structures, or other means.
- Experimental evidence that terrain datasets routinely contain a very few well defined highest visibility points is important to the manner in which solutions to certain types of siting problems are undertaken.
- A specific elevation model and visibility characteristics taxonomy is presented to enable the results of this research to be understood and extended in a manner that facilitates further development and meaningful comparisons with other work.

Several issues and goals relating to this work were described in Chapter 1. They have been addressed as follows:

1. We use all relevant data from the elevation raster. No points are dropped by simplifying the dataset, and the criterion for adequacy and criterion for appropriateness are satisfied.
2. Significant improvements in performance over existing implemented and accurate algorithms have been achieved for total viewshed determination by R2 in  $\Theta(R^2)$  time and for visibility index estimation by WeightF in  $\Theta(R)$  time.

3. The accuracy and utility of the algorithms developed have been confirmed on examples of different types of terrain (mountain, river valley, marshland) and for different LOS parameters (antenna, human vision).
4. Visibility indexes on real terrain have been observed to follow distributions resembling negative exponential or log-normal forms; a very limited number of points with high visibility index values for their region can be identified and used as input to various site selection algorithms.
5. Parallelism at various degrees of granularity and architectural dispersion have been successfully employed to tackle visibility analyses with large, real world problem sizes. This exploitation of parallel execution included robust recovery and input data sharing.
6. The algorithms developed have been shown to be practical and implementable; demonstration viewers for inexpensive platforms have been made available to user communities to confirm the utility of visibility index images and to identify areas for further research. The utility of these algorithms have been demonstrated on 1 meter spacing elevation datasets.

Any existing algorithms or applications that make use of visibility rankings or viewsheds can potentially benefit from using faster visibility determination methods. GIS-related applications are obvious instances. Use of R2 instead of R3 in an on-line viewshed display application reduces the cost of visibility computation to the same order as for screen pixel update operations. Outside of the traditional GIS area, the visibility image applications described in [DFJN92] and [Ray94] benefit from total scene analysis times reduced from  $\Theta(R^5)$  using naive methods to  $\Theta(R^4)$  by using R2 and (if accurate enough in the specific problem domain) to  $\Theta(R^3)$  using WeightF.

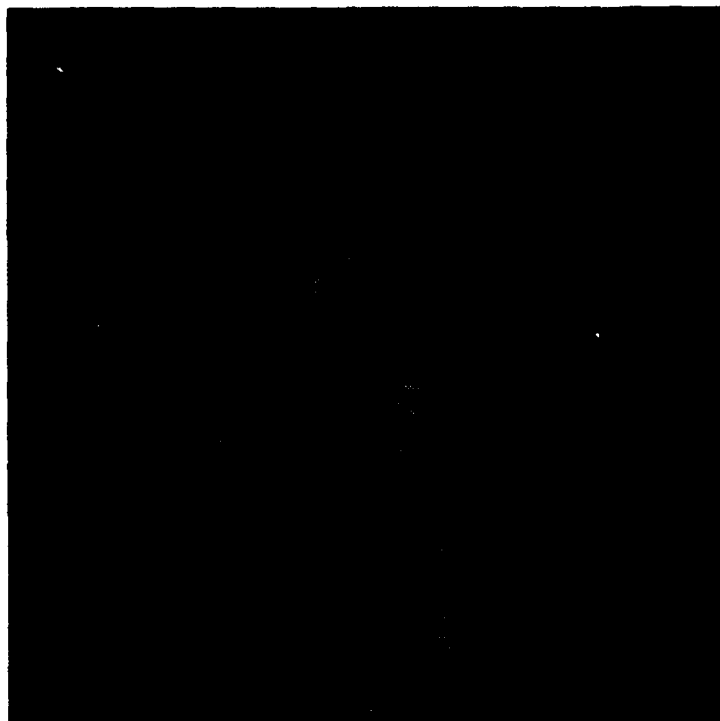
Given the existence of a set of portable, reusable, and extensible implementations of algorithms to conduct fast and accurate visibility analysis of large elevation

datasets, a variety of new approaches to existing problems can be undertaken. Computationally cheap and robust visibility calculations can open new possibilities for analyzing the occurrences and nature of errors in elevation datasets. In addition to the analytical possibilities (such as the impact of elevation data errors on visibility in Figure 6.28), new visual aids in error identification become possible.

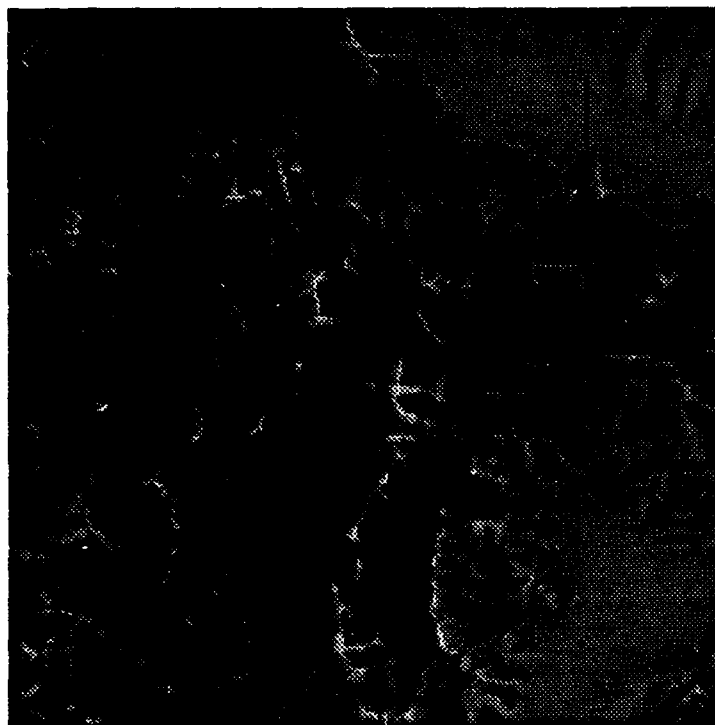
Figures 8.1 and 8.2 show an instance of such an application. They show the elevation and visibility gray-scale images of a portion of the terrain in the southeast corner of DTED cell N37E126. Note the dark vertical line in the northeast corner of the visibility image. This line is barely distinguishable in the elevation image, even knowing where to look for it, and is virtually undetectable to the unaided eye on a standard gray-scale workstation display of elevation data. The line corresponds to an erroneous drop of 10–20 meters in the elevation values recorded on the boundary of a DTED cell. This appears to be another instance of the nature and magnitude of DEM strip error mentioned in Section 2.3. It is difficult to identify in the elevation image, but the visibility values for the corresponding points in the visibility image decrease in intensity by 60% to 80% and show the problem area clearly.

### 8.3 Future Work

- *Faster Real Terrain Visibility Methods* Tools already created can facilitate the investigation of even faster visibility determination algorithms. Such methods could capitalize on correlated viewshed subregions between adjacent points, terrain configuration local to an observer point, genetic algorithms, neural networks, quadtree representations of minimum and maximum elevation values within subregions, and various other approaches. The ultimate objective of such investigations would be to produce algorithms that run significantly faster than WeightF and R2 but yield the same relative degrees of accuracy (or better).



**Figure 8.1: Elevation Vicinity n37e126y**



**Figure 8.2: Visibility Vicinity n37e126y**



- *Variations in Elevation Point Density* Given the availability of reliable elevation datasets of varying densities for the same area, the relative impact of random error and grid density on visibility results can be investigated. Simple elimination of points from an existing grid may not prove suitable, since an agent producing a reduced density DEM for real applications would be likely to take into account issues of 'representativeness' and apply some type of adjustment to the chosen values.
- *Terrain Complexity Categorization* Computationally efficient means of categorizing terrain complexity can be evaluated by observing the relationship between their measures and the number of WeightF LOS rays needed to obtain a specified degree of visibility accuracy for given terrain data sets. The general concept of using the emergence of patterns between GIS operations as a means of identifying valid and properly scaled measurement methods is mentioned in [Ope89]. Existing methods such as the 2D-Fourier Transform and Fractal Dimension Estimation are computationally intensive, and simple statistics (e.g., elevation standard deviation) are insufficient to distinguish between terrain types such as gradually sloping plains and rolling hills. Possible measures to investigate include the number of rays required for accurate WeightF estimates, slope sign changes along rows or columns of elevation data, and spatial correlation of elevation data.
- *Relationship of Aspect Error to Visibility* As mentioned in Section 2.3, random DEM errors may have significant impacts on the apparent directional aspect of a terrain feature represented in a DEM. Aspect errors may result in significant visibility errors for some terrain features by narrowing or expanding their potential field of view around obstructions. The algorithms developed during this research can be used to help investigate variations in

visibility strongly related to error-induced aspect variations. The visibility images described in Section 5.7.2 can aid in visualizing aspect and visibility relationships.

Other areas for extensions and future research include investigation of other terrain areas (including those on Venus or Mars by using data from NASA), incremental application of WeightF calculations for early identification of best visibility points, modeling memory cache behavior during visibility calculations on large viewsheds, and weighting point contributions to viewsheds using the calculated LOS height at the point.

## LIST OF SYMBOLS

$S_c$	The controlling slope at any specific state in the process of a visibility determination.
$B$	The set of target points on the boundary of an area viewed from a given observer.
$C$	A class of observed point associated with a specific weight vector for the value of covering it by various numbers of observer facilities.
$D_O$	Distance from the observer in earth curvature calculations.
$\Delta E_C$	Change in elevation due to earth curvature.
$GD[p_a, p_b]$	The ground distance between points $P_a$ and $P_b$ in $L$ .
$G_a$	One specific arrangement of facilities within $S_f$ .
$L$	The surface of the landscape of an AOI.
$M$	A matrix of points corresponding to $T$ .
$O_h$	The height of an observer above ground level.
$P_a$	A point in 3D space on or above $L$ .
$R$	The maximum range for which visibility is significant.
$R_{ave}$	The average range for which visibility is significant.
$R_E$	Effective radius of the earth.
$S$	A $2\frac{1}{2}$ D surface where elevation is defined only for locations corresponding to grid points or grid lines in $M$ .
$S_f$	Set of points in $S$ where it is possible to place an observer facility.
$S_{xy}$	The set of $(x, y)$ coordinates in $S$ for which elevation is defined.
$T$	A tessellation of integer valued elevations which approximates $L$ .
$V_{ab}$	Visibility function of observer at $P_a$ and target at $P_b$ .
$c_i$	The class of the point at location $i$ in $S$ .
$c_{ia}$	The class of the point at location $i$ using facility arrangement $a$ .
$f_L$	A function mapping a 2D geographic coordinate to the elevation of $L$ at that point.
$d_T$	The error in the approximation of $L$ by $T$ at a given point.
$h_t$	The lowest height above ground for which it is necessary to establish a LOS to a target.
$m_{sz}$	Constant terrain distance between all longitudinally adjacent points in $M$ .

$m_{xy}$	Constant terrain distance between all latitudinally adjacent points in $M$ .
$n_C$	Number of classes of covered points in $S$ .
$n_f$	Number of observer facilities available.
$n_G$	The number of possible distinct arrangements of facilities.
$n_M$	The number of points in $M$ .
$n_{IC}$	The number of points improperly contributing to a LOS.
$n_{PC}$	The number of points properly contributing to a LOS.
$n_R$	The average number of grid point or grid segment crossings along an LOS of length $R$ .
$n_S$	The number of elevation data points in $S$ .
$n_{S_f}$	The number of potential observer site points in $S_f$ .
$n_{VI}$	The number of LOS rays used in calculating a visibility index estimate.
$p_a$	A point in $M$ corresponding to a point $P_a$ in $L$ .
$w_{ki}$	Weight (value) of covering a point in class $C_k$ by $i$ facilities.
$z_L$	The elevation of a point in $L$ .
$z_S$	The elevation of a point in $S$ where elevation is defined.
$\Theta()$	notation bounds a function to within constant factors.
$O()$	notation gives an upper bound for a function to within a constant factor.
$\Omega()$	notation gives a lower bound for a function to within a constant factor.

## GLOSSARY

**4-Neighbor Adjacent** Relationship between two points that are immediately north-south or east-west adjacent on a grid.

**8-Neighbor Adjacent** Relationship between two points that are immediately north-south, east-west, northwest-southeast, or northeast-southwest adjacent on a grid.

**ALBE** Air-Land Battle Environment. An evolving set of terrain analysis applications sponsored by TEC which serves as a testbed for new concepts.

**AM** Altitude Matrix. A regular rectangular grid of point measures that forms the most common type of Digital Elevation Model [Wal89].

**AOI** Area of Interest. The total area which must be represented to evaluate a potential siting solution for a given situation.

**Controlling Slope** With respect to a given observer, the slope of a line segment beneath which no point is visible on or over the terrain for which that slope is the controlling slope.

**Covering Triangle** A triangle defined by two grid line segments with one common endpoint which is assumed to be no lower than any corresponding terrain points.

**DEM** Digital Elevation Matrix.

**DMA** Defense Mapping Agency.

**DTED** Digital Terrain Elevation Data produced by DMA, typically in cells representing an area of one degree of latitude by one degree of longitude. DTED

Level I data has a typical grid spacing of three-arc-seconds, while typical DTED Level II data has grid spacing of one-arc-second.

**FSO** Field Support Office, TEC.

**GIS** Geographic Information System.

**GPS** Global Positioning System. A system of orbiting satellites and associated standards for receiving their signals which allows determination of relatively precise 3D locations anywhere on the surface of the earth.

**HAWK** A mobile medium-capability air defense missile and radar system capable of tracking and engaging targets within a 360° circle of its emplacement site subject to minimum and maximum target range and altitude constraints.

**KB** Kilobytes — approximately 1 thousand bytes.

**LAN** Local Area Network. A network connecting stations residing in essentially the same geographic location. Throughput is typically on the order of 1 MB per second (max) delivered to each station.

**LOS** Line-of-Sight.

**LOS Height** The lowest height at which a target at a given location is visible from the observer.

**MAUP** Modifiable Areal Unit Problem. Affects on the values, relationships, and inferences derived from spatial data due to the number and size of the zones used to report the data.

**MB** Megabytes — approximately 1 million bytes.

**NFS** Network File System.

**Radius of Visibility** The maximum geographic distance (meters or kilometers) for which visibility from the observer is significant. This might be the maximum range of a radar, or of optical visibility under prevailing weather conditions.

**RMSE** Root Mean Squared Error.

**SPECmark** A relatively standard measure of CPU performance between different processors.

**TEC** U.S. Army Topographic Engineering Center.

**TerraBase** A terrain database system for MS-DOS based personal computers originally developed at the U.S. Military Academy and now supported at the FSO.

**TIN** Triangulated Irregular Network.

**USGS** United States Geological Survey.

**Viewshed** The points within the radius of visibility that are visible from the observer.

**Visibility Fraction** The ratio of the number of points in the viewshed to the total number of points within the radius of visibility.

**Visibility Index** A single numeric measure of the amount of terrain which is visible from a particular point compared with other possible observer points in the same dataset.

**WAN** Wide Area Network. A network connecting stations that are geographically distributed, generally with throughput an an order of magnitude less than found on LANs.

## REFERENCES

- [Bit92] Wolfgang Bitterlich. Preserving and representing key landscape elements in a GIS. In *SORSA 1992 Workshop and Symposium Proceedings*, pages 74-91, July 1992.
- [Bro80] Barbara D. Broome. SEEFAR: An improved model for producing line-of-sight maps. Technical Report 225, Aberdeen Proving Grounds, September 1980.
- [BSPD90] Barbara G. Beckerman, John L. Smyre, Stephen L. Packard, and Richard C. Durfee. Survey and analysis of object-oriented, real-time, three-dimensional terrain visualization and analysis capabilities. Technical Report K/DSRD-588, Computing and Telecommunications Division, Oak Ridge National Laboratory, December 1990.
- [Car90] Lieutenant Colonel William L. Carr. Stating air defense tasks and missions. Individual study project, U.S. Army War College, Carlisle Barracks, Pennsylvania, May 1990.
- [CdB72] S. D. Conte and Carl de Boor. *Elementary Numerical Analysis*. McGraw-Hill Book Company, 1972.
- [CG93] Douglas R. Caldwell and Linda H. Graff. Directional regions in geographic information systems. In *GIS/LIS Proceedings*, volume 1, pages 83-91, November 1993.
- [CLR91] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Company, 1991.
- [CMS90] John Current, Hokey Min, and David Schilling. Multiobjective analysis of facility location decisions. *European Journal of Operational Research*, 49(3):295-307, December 1990.
- [Com88] Computer Graphics Lab, Department of Geography and Computer Science, West Point, New York. *TERRABASE User Guide*, 1988.
- [CR74] Richard Church and Charles ReVelle. The maximal covering location problem. In *The Regional Science Association Papers*, pages 101-118, 1974.
- [DD93] Catherine Dibble and Paul J. Densham. Generating interesting alternatives in GIS and SDSS using genetic algorithms. In *GIS/LIS Proceedings*, volume 1, pages 180-189, November 1993.



- [Den93] Paul J. Densham. Integrating GIS and parallel processing to provide decision support for hierarchical location selection problems. In *GIS/LIS Proceedings*, volume 1, pages 170–179, November 1993.
- [DFFP+86] Leila De Floriani, Bianca Falcidieno, Caterina Pienovi, David Allen, and George Nagy. A visibility-based model for terrain features. In *Proceedings 2nd International Symposium on Spatial Data Handling*, pages 235–250, July 1986.
- [DFJN92] Leila De Floriani, Philippe Jeanne, and George Nagy. Visibility-related image features. *Pattern Recognition Letters*, 13(6):463–470, June 1992.
- [DFN88] Leila De Floriani and George Nagy. Visibility problems on a topographic surface. Technical Report 59, Istituto per la Matematica Applicata, Genova, 1988.
- [DFNN88] Leila De Floriani, George Nagy, and Hari Nair. A visibility-based model for terrain features. Technical Report 88-824, Computational Geometry Laboratory, Rensselaer Polytechnic Institute, November 1988.
- [Dut89] Geoffrey Dutton. Modeling locational uncertainty via heirarchical tessellation. In Michael Goodchild and Sucharita Gopal, editors, *The Accuracy of Spatial Databases*, chapter 12. Taylor & Francis, 1989.
- [Eyt89] J. Ronald Eyton. Digital landform models: A numerical and pictorial atlas. Technical report, Department of Geography, The University of Alberta, 1989.
- [Fis91] Peter F. Fisher. Modeling and visualizing uncertainty in geographic data. In *National Center for Geographic Information and Analysis: Specialist Meeting*, June 1991.
- [Fis92] Peter F. Fisher. First experiments in viewshed uncertainty: Simulating fuzzy viewsheds. *Photogrammetric Engineering & Remote Sensing*, 58(3):345–352, March 1992.
- [Fis93] Peter F. Fisher. Algorithm and implementation uncertainty in viewshed analysis. *International Journal of Geographical Information Systems*, 7(4):331–347, 1993.
- [Fot89] A. Stewart Fotheringham. Scale-independent spatial analysis. In Michael Goodchild and Sucharita Gopal, editors, *The Accuracy of Spatial Databases*, chapter 19. Taylor & Francis, 1989.

- [FR94a] Wm Randolph Franklin and Clark Ray. Higher isn't necessarily better: Visibility algorithms and experiments. In preparation, Rensselaer Polytechnic Institute, 1994.
- [FR94b] Wm Randolph Franklin and Clark Ray. A new perspective on visibility. In preparation, 1994.
- [Fra92] Wm Randolph Franklin. Map overlay area animation and parallel simulation. In *SORSA 1992 Workshop and Symposium Proceedings*, pages 200-202, July 1992.
- [GCR92] L.D. Griffin, A. C. F. Colchester, and G. P. Robinson. Scale and segmentation of grey-level images using maximum gradient paths. *Image & Vision Computing*, 10(6):389-402, 1992.
- [Gie93] James M. Giesken. The shift toward digital geographic information for air combat operations. In *GIS/LIS Proceedings*, volume 1, pages 246-252, November 1993.
- [GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability*. Bell Telephone Laboratories, Inc., 1979.
- [Goo91] Michael F. Goodchild. Position paper for I7 specialist meeting. In *National Center for Geographic Information and Analysis: Specialist Meeting*, June 1991.
- [Goo92a] M. F. Goodchild. Dealing with uncertainty in spatial data. Workshop, Symposium on Spatial Data Handling, August 1992.
- [Goo92b] Michael F. Goodchild. Geographical data modeling. *Computers & Geosciences*, 18(4):401-408, 1992.
- [GS86] Bruce L. Golden and Christopher C. Skiscim. Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly*, 33:261-279, 1986.
- [HB92] Fred Holroyd and Sarah B. M. Bell. Raster GIS: Models of raster encoding. *Computers & Geosciences*, 18(4):419-426, 1992.
- [Jan89] Jane's Information Group, Alexandria, Virginia. *Jane's Weapon Systems*, 1988-1989.
- [Jan90] Jane's. Medium-range surface-to-air missiles. *International Defense Review*, 23:969, September 1990.
- [Jun89] D-M. Jung. Comparisons between algorithms for terrain visibility. Master's thesis, Rensselaer Polytechnic Institute, 1989.

- [Kna92] Wim G. M. Van Der Knapp. The vector to raster conversion: (mis)use in geographical information systems. *International Journal of Geographical Information Systems*, 6(2):159-170, 1992.
- [Knu81] Donald E. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley Publishing Company, 2 edition, 1981.
- [Kum92] Mark P. Kumler. *An Intensive Comparison of TINs and DEMs*. PhD thesis, University of California, Santa Barbara, 1992.
- [Lay93] Jinn-Guey Lay. Experiments on error of digital elevation models. In *GIS/LIS Proceedings*, volume 1, pages 389-397, November 1993.
- [Lee91a] Jay Lee. Analyses of visibility sites on topographic surfaces. *International Journal of Geographical Information Systems*, 5(4):413-429, 1991.
- [Lee91b] Jay Lee. Comparison of existing methods for building triangular irregular network models of terrain from grid digital elevation models. *International Journal of Geographical Information Systems*, 5(3):267-285, 1991.
- [Lee92] J. Lee. Visibility dominance and topographic features on digital elevation models. In *Proceedings 5th International Symposium on Spatial Data Handling*, volume 2, pages 622-631, August 1992.
- [Li93] Bin Li. Developing network-oriented GIS software for parallel computing. In *GIS/LIS Proceedings*, volume 1, pages 403-413, November 1993.
- [Mil93] Harvey J. Miller. GIS and the geometry of facility location problems. In *GIS/LIS Proceedings*, volume 2, pages 530-539, November 1993.
- [MM85] Juan A. Cebrian James E. Mower and David M. Mark. Analysis and display of digital elevation models within a quadtree-based geographic information system. In *Auto-Carto 7 Proceedings*, pages 55-64, March 1985.
- [Mow93] James E. Mower. Implementing GIS procedures on parallel computers: A case study. In *Auto-Carto 11 Proceedings*, pages 424-433, October 1993.
- [Nag94] George Nagy. Terrain visibility. In preparation, Rensselaer Polytechnic Institute, 1994.
- [Nai88] Hari Nair. A high-level description of digital terrain models using visibility information. Master's thesis, Rensselaer Polytechnic Institute, 1988.

- [Ope89] Stan Openshaw. Learning to live with errors in spatial databases. In Michael Goodchild and Sucharita Gopal, editors, *The Accuracy of Spatial Databases*, chapter 23. Taylor & Francis, 1989.
- [Pal91] James F. Palmer. Representing error in GIS modeling. In *National Center for Geographic Information and Analysis: Specialist Meeting*, June 1991.
- [PDDT92] E. Puppo, L. Davis, D. DeMenthon, and Y. A. Teng. Parallel terrain triangulation. In *Proceedings 5th International Symposium on Spatial Data Handling*, volume 2, pages 632-641, August 1992.
- [PM93] Michael F. Polis and Jr. David M. McKeown. Issues in iterative TIN generation to support large scale simulations. In *Auto-Carto 11 Proceedings*, pages 267-277, October 1993.
- [Rav93] Balasubramaniam Ravichandran. *2D and 3D Model-Based Matching Using a Minimum Representation Criterion and a Hybrid Genetic Algorithm*. PhD thesis, Rensselaer Polytechnic Institute, May 1993.
- [Ray94] Clark K. Ray. A new way to see terrain. Accepted for publication in *Military Review*, 1994.
- [RFM92] Clark K. Ray, William Randolph Franklin, and Shashank Mehta. Geometric algorithms for siting of air defense missile batteries. Technical Report 2756, Rensselaer Polytechnic Institute, 1992.
- [RR92] Sanguthevar Rajasekaran and John H. Reif. Nested annealing: a provable improvement to simulated annealing. *Theoretical Computer Science*, (99):157-176, 1992.
- [Sha90] A. Shapira. Visibility and terrain labeling. Master's thesis, Rensselaer Polytechnic Institute, 1990.
- [SI71] Robert E. Shannon and James P. Ignizio. Minimum altitude visibility diagram - MAVD. *Simulation*, pages 256-260, June 1971.
- [Slo93] Kevin R. Slocum. Conifer tree influence on digital terrain elevation data: A case study at Dulles International Airport. Technical Report TEC-0038, U.S. Army Corps of Engineers Topographic Engineering Center, Fort Belvoir, Virginia 22060-5546, September 1993.
- [TD92] Y. Ansel Teng and Larry S. Davis. Visibility analysis on digital terrain models and its parallel implementation. Technical Report CAR-TR-625, Center for Automation Research, University of Maryland, College Park, Maryland, May 1992.

- [TDD91] Y. Ansel Teng, Daniel DeMenthon, and Larry S. Davis. Region-to-region visibility analysis using massively parallel hypercube machines. Technical Report CAR-TR-578, Center for Automation Research, University of Maryland, College Park, Maryland, September 1991.
- [TM75] J. P. Tremblay and R. Manohar. *Discrete Mathematical Structures with Applications to Computer Science*. McGraw-Hill Book Company, 1975.
- [TR72] Constantine Toregas and Charles ReVelle. Optimal location under time or distance constraints. In Morgan D. Thomas and Joan B. Manzer, editors, *The Regional Science Association Papers*, volume 28, pages 133-143. Regional Science Association, 1972.
- [TR74] Constantine Toregas and Charles ReVelle. The maximal covering location problem. In Morgan D. Thomas and Joan B. Manzer, editors, *The Regional Science Association Papers*, volume 32, pages 101-118. Regional Science Association, 1974.
- [Tsa93] Victor J. D. Tsai. Delaunay triangulations in TIN creation: an overview and a linear-time algorithm. *International Journal of Geographical Information Systems*, 7(6):501-524, 1993.
- [TV93] Victor J. D. Tsai and Alan P. Vonderohe. Delaunay tetrahedral data modelling for 3-d GIS applications. In *GIS/LIS Proceedings*, volume 2, pages 671-680, November 1993.
- [USG90] USGS. Digital elevation models—data users guide. National Mapping Program Technical Instructions 5, Office of Technical Management, USGS, Reston, Virginia, 1990.
- [Wal89] Stephen J. Walsh. User considerations in landscape characterization. In Michael Goodchild and Sucharita Gopal, editors, *The Accuracy of Spatial Databases*, chapter 3. Taylor & Francis, 1989.
- [WF93] Joseph D. Wood and Peter F. Fisher. Assessing interpolation accuracy in elevation models. *IEEE Computer Graphics & Applications*, 13(2):48-56, March 1993.
- [YB93] Bruce J. Yankowiak and John C. Bohlke. Predicting the accuracies of GPS derived elevations using a gravity map of Minnesota. In *GIS/LIS Proceedings*, volume 2, pages 811-819, November 1993.
- [YM92] Peter J. Young and David W. Mason. Digital terrain requirements for low level combat simulations. In *SORSA 1992 Workshop and Symposium Proceedings*, pages 112-121, July 1992.

## APPENDIX A

### Computation and Storage Costs of Visibility Determinations

As discussed elsewhere, the general case of establishing a LOS from an observer to all surrounding grid elevation points requires running a unique evaluation to each point for which an existing collinear LOS (to a previous point) cannot be extended. The actual execution time on a Sun IPC workstation with a performance level of approximately 11 SPECmarks provided a mean throughput of 256 point-to-point LOS evaluations per second with a maximum radius of visibility of 40km in DTED cell N37E127. This distance was selected from [Jan89] and [Jan90] as representative of existing and projected medium capability air defense radar ranges.

A standard DTED Level I cell represents a rectangular area approximately 100km on a side, which is a realistic magnitude for communications or radar planning. The grid for such a cell is typically  $1201 \times 1201$  for a total of 1,442,401 elevation points. Given a three-arc-second grid spacing in each cell, the ground distance (North-South and East-West) between points varies between 60 and 100 meters, depending on the specific latitude. The rest of this analysis uses an average figure of approximately 80 meters (this matches the 1200 grid spaces per side of a DTED Level I cell covering approximately 100km). Since each elevation point is measured in meters, a two-byte integer (which can represent values between 32,767 to -32,768 meters of altitude from mean sea level) is a common storage unit for elevations. The total storage requirements to represent a  $10,000\text{km}^2$  area is therefore approximately 3MB. However, the need to include surrounding areas out to the desired radius of visibility requires additional storage. Extending data coverage 40km in each direction requires a grid of approximately  $2200 \times 2200$  points for a storage requirement of approximately 10MB.

A program to calculate visibility values on the Sun IPC was coded in the

G++ implementation of the C++ programming language, using all-integer operations in critical program segments. Actual throughput for 40km LOS evaluations was approximately 128 evaluations per CPU second. When evaluating LOS's to *each* point on a fixed number of LOS rays within a 40km radius of visibility, the average distance will be 20km, implying a throughput of approximately 256 LOS evaluations per CPU second. Allowing for an improvement by a factor of four in some other common environment (more powerful hardware, better optimizing compiler, etc.), a throughput of approximately 1000 LOS evaluations per second under the conditions stated above seems reasonably obtainable, implying 1 millisecond of CPU time per LOS evaluation.

The computational cost to evaluate a given LOS will vary linearly with the density of grid points to ground distance. This is because determining the LOS between two points requires values (or estimates) of elevation based on all elevation data points in between. If the cost to calculate a LOS between two points is 0.001 seconds when the average grid spacing is 80 meters, it would take approximately 0.002 seconds if the average grid spacing were 40 meters.

The estimated CPU time to evaluate the LOS from each point in a 10000km<sup>2</sup> area to each point on the elevation data grid within a 40km radius of visibility in terms of the ground spacing between grid points can be calculated from the following relations:

$$A_{km} = (100km)^2$$

$$N_{km} = 1000m/S_g$$

$$N_o = A_{km}/N_{km}^2$$

$$N_c = (\pi \times R_v^2)/S_g^2$$

$$T_{los} = (0.08sec/m)/S_g \quad (A.1)$$

$$T_{tot} = N_o \times N_c \times T_{los} \quad (A.2)$$

where

$A_{km}$  Area containing potential observer points, in  $km^2$ .

$N_{km}$  Number of observer points per linear  $km$ .

$N_c$  Number (approximate) of target points in a circle of radius  $R_v$  meters from a given observer point.

$N_o$  Number of observer points.

$R_v$  Radius of visibility, 40,000 meters in this case.

$S_g$  Average ground spacing between points, in meters.

$T_{los}$  Time (mean) to evaluate a single 20km LOS, estimated as 0.001 seconds when  $S_g = 80$  meters.

$T_{tot}$  Total time to evaluate a LOS to each target point from each observer point, in seconds.

If  $D_g$  is defined as the average linear density of grid points per kilometer ( $1000m/S_g$ ), then from the above relations an expression for total time is found to be:

$$T_{tot} = (100km)^2 \times D_g^2 \times \pi \times (40km)^2 \times D_g^2 \times 0.00008 \times D_g$$

This reduces to:

$$T_{tot} = (1280\pi)D_g^5 \quad (A.3)$$

so clearly

$$T_{tot} = \Theta(D_g^5) = O(D_g^5) = \Omega(D_g^5)$$

for this particular visibility algorithm, where [CLR91]:

- $\Theta$  notation bounds a function to within constant factors.
- $O$  notation gives an upper bound for a function to within a constant factor.



- $\Omega$  notation gives a lower bound for a function to within a constant factor.

Therefore, the total time to evaluate the LOS from each potential observer point to all corresponding target points will vary with the linear data point density raised to the fifth power.

## APPENDIX B

### Formal Definition of the HAWK Siting Problem

The variety of approaches to the problems of visibility determination and facility site selection have naturally led to different forms of notation in describing specific problem domains. The following description attempts to make use of aspects common to several existing definitions ([DFN88], [Lee91a], [SI71], [CR74]). The model of the elevation data and its relationship to the physical terrain setting are described in Chapter 3.

Consider a surface  $S$  which contains a total of  $n_S$  elevation data points and corresponds to the landscape  $L$  that contains the AOI for a given siting problem. Let  $h_o$  be the effective height of the observer (in this case a HAWK radar antenna) above the ground and let  $h_t$  be the minimum height above ground level at which it is deemed necessary to track a target. To be able to accurately track (as opposed to just detect) an aerial target, the target must be *visible* from the observer. In this model points  $P_a$  and  $P_b$  in three-dimensional space are each visible from the other if a straight line between them is not beneath the surface  $S$  at any point where the elevation  $z_S$  is defined. Stated differently, if for any point  $(x_l, y_l, z_l)$  on the line connecting  $P_a$  and  $P_b$  where  $(x_l, y_l) \in S_{xy}$  it is true that  $z_l \geq z_S$  then  $P_a$  is visible from  $P_b$  and  $P_b$  is visible from  $P_a$ .

The symmetry in visibility between  $P_a$  and  $P_b$  in general three-dimensional space no longer holds when we consider points which are the projections onto  $S$  of potential observer and target locations. If an observer  $O_a$  is positioned at point  $(x_a, y_a, z_a)$ , where  $(x_a, y_a) \in S_{xy}$  and  $z_a = f_S(x_a, y_a)$ , then for visibility determination we consider lines-of-sight to originate at  $(x_a, y_a, (z_a + h_o))$  and extend out to  $(x_b, y_b, (z_b + h_t))$  where  $(x_b, y_b) \in S_{xy}$ . Since in general  $h_o \neq h_t$ , if an observer at  $P_a$  can see a target at  $P_b$  this does not imply that an observer at  $P_b$  could see a target

at  $P_a$ . Define the visibility function  $V_{ab}$  to have the value 1 if a target no lower than height  $h_t$  above point  $P_b$  on surface  $S$  is visible from an observer at height  $h_o$  above point  $P_a$  on surface  $S$  and the distance between  $P_a$  and  $P_b$  is not greater than some constant  $R$ . Let  $n_R$  be the average number of crossings corresponding to points in  $M$  (see Chapter 3) where a LOS of length  $R$  crosses a grid line segment.

$V_{ab}$  has a value of 0 otherwise. Note that the value of  $V_{ab}$  implies nothing about the value of  $V_{ba}$ . If  $V_{ab} = 1$  we say that point  $P_b$  can be *covered* from an observer at point  $P_a$ .

Let  $n_f$  be the number of facilities (HAWK units) available. The number of units ( $0 \dots n_f$ ) covering the various points in  $S$  determines the effectiveness of a particular arrangement of HAWK units in an AOI. For each location  $i \in S$ , where  $i = 1, 2, \dots, n_S$  let  $f_i = 1$  if there is a facility at location  $i$  and  $f_i = 0$  otherwise. The number of facilities at any given location is always either 0 or 1.

Not all points in  $S$  will have the same importance for coverage. Each point in  $S$  is considered to belong to exactly one of  $n_C$  classes for the purposes of evaluating the value (weighting) of covering it. Let  $c_i$  be the class of the point at location  $i$ . Within each class  $C_k, k = 1 \dots n_C$ , a different weight exists which corresponds to the number of facilities ( $0 \dots n_f$ ) covering a point in that class. Let  $w_{kj}$  be the weight assigned to the value of covering a single point in class  $C_k$  by exactly  $j$  facilities. We impose no particular restrictions on the values chosen for these weights other than  $-\infty \leq w \leq \infty$ , although a typical scenario would have  $0 \leq w < \infty$  with the value of  $w_{kj}$  increasing as  $j$  increases.

Let  $S_f$  be the set of points in  $S$  where it is possible to place a facility, and let  $n_{S_f}$  be the number of such points.  $n_{S_f} \leq n_S$ , and generally  $n_{S_f} = \Omega(n_S)$  over various AOI's. If  $k \in S_f$  then  $f_k = 1$ . In typical real world situations  $n_f \ll n_{S_f}$ . Let  $G_a$  be an arrangement of facilities,  $a \in 1, \dots, n_G$ , such that for each arrangement there are exactly  $n_f$  locations  $i \in S_f$  for which  $f_i = 1$ . There will be  $n_G$  possible

distinct arrangements, where

$$n_G = \frac{n_{S_f}!}{n_f!(n_{S_f} - n_f)!} \quad (\text{B.1})$$

For a specific problem with a fixed number of available facilities  $n_f = k$  and where  $n_f \ll n_{S_f}$ , then  $n_f! = k!$  and we can approximate the variation in  $n_G$  with  $n_{S_f}$  as

$$n_G \approx (n_{S_f})^{n_f} \quad (\text{B.2})$$

Observe that if the linear density of points on the ground (see Table 1.1) doubles, then the number of possible arrangements can be expected to increase by a factor of approximately  $4^k$ .

We can now state the initial, or *static weight* HAWK siting problem as finding an arrangement  $G_a$  which maximizes

$$\sum_{i \in S} w_{c_i b_i} \quad (\text{B.3})$$

where

$$b_j = \sum_{p \in G_a} V_{pj}$$

for locations  $p \in G_a$ . Note that this formulation implicitly assumes that there is no change in weights between different arrangements, and therefore no value is imputed to coverage of one facility by another.

Since in the real world this is often not the case (air defense units are sited to cover each other, when possible), an alternative formulation to represent the value of overlapping coverage becomes finding an arrangement  $G_a$  which maximizes

$$\sum_{i \in S} w_{c_{ia} b_i} \quad (\text{B.4})$$

where  $c_{ia}$  is the class to which location  $i$  belongs under arrangement  $a$ . This problem is referred to as the *dynamic weight* HAWK siting problem. The class of a given location will vary depending on whether coverage of that location is considered significant in defending one or more HAWK sites in any given arrangement.

It is useful at this point to emphasize that it is necessary to perform the summations in either problem formulation for all locations  $i \in S$ , not just locations where specific targets (or HAWK sites, in the dynamic weight formulation) are located. This is due to need to intercept or deter potential attackers before they have approached close enough to their targets to release stand-off munitions, which have a range of several kilometers or more. The need to evaluate the contribution of all covered points in  $S$  also conforms well to the problems related to siting communications facilities and other applications influenced by visibility considerations.

## APPENDIX C

### Investigation of Elevation-Visibility Relationships

A variety of factors influence the degree of correlation observed between elevation and visibility at given points. These include but are not limited to the roughness and extent of the sample area, presence of significant coastline and sea level areas, considering all or only selected sets of points in the sample area, the model used to relate elevation and visibility, and the method used to calculate the visibility index for observer points. The results of some of these correlation calculations are presented here in tabular form for two models. The linear model assumes that for the area sampled

$$V_i = \beta_0 + \beta_1 z_i + \epsilon_i \quad (\text{C.1})$$

while the quadratic model assumes

$$V_i = \beta_0 + \beta_1 z_i^2 + \epsilon_i \quad (\text{C.2})$$

where  $V_i$  is the visibility index at location  $i$ ,  $z_i$  is the elevation at location  $i$ ,  $\beta_0$  and  $\beta_1$  are parameters, and  $\epsilon_i$  is the error between the predicted and actual visibility index at location  $i$ . The empirical basis for not including a linear term in Equation C.2 is presented in Chapter 6. The most significant results of the regression analysis are the sign and magnitude of the correlation coefficient  $r$ . The value of  $r^2$  (the correlated variation) reflects the portion of the variation in the value of  $V_i$  that is explained by the assumed relationship to  $z_i$ . Many other models relating elevation and visibility can be constructed; here we attempt only to examine the results of these two simple models over a variety of terrain areas for which elevation data is available and visibility index values have been computed.

Results obtained by applying the two models described above to several DTED cells are shown in Table C.1. Points with elevation values of zero (i.e., sea level) were

Correlation of Visibility to Elevation: Best Points						
Cell Name	Non-0 Elevs	Vis Index Range	Best Points	r Linear	r Quadratic	Cell Type
N37E127	1441304	144-16976	2071	0.240055	0.161028	Mountain Lowland
N36E127	1439567	160-19800	9980	0.271448	0.201289	Mountain Lowland
N36E128	1442399	160-16648	690	0.20519	0.176861	Mountain Upland
N36E126	759014	184-31704	1529	-0.024374	-0.052767	Coastal
N37E127-W	1441304	8-14624	582	0.068689	0.018259	Mountain Lowland

**Table C.1: Best Points Visibility—Elevation Correlations.**

excluded from the correlation calculations, but not in the calculation of the visibility index values for points that are considered. As mentioned elsewhere, the potential values for a visibility index for a given point may range from 0 (perhaps from the bottom of a well!) to 32767 (all points within radius  $R$  are visible). The number of observations used in calculating correlations varies with the number of sea level points excluded. N36E128 has almost no points excluded (from a maximum possible of 1,442,401), while N36E126 is almost one-half ocean and has almost 50% of its points excluded. The '-W' in the cell name N37e127-W indicates that in computing the visibility index values, the contribution of each point along each of the 32 LOS's calculated was *weighted* approximately according to its distance from the observer point. This variant is included to evaluate the impact of treating each point along a LOS as representing a sample of all unevaluated points closer to it than to any other evaluated (i.e., on a LOS) point.

Several significant trends can be noted in Table C.1. First, the number of points with visibility values in the top half of the total range of values for a given cell is much smaller than the total points in the cell. For example, for N37E127 out of 1,441,304 non-sea level points in the cell, only 2071 had visibility index values

in the top half (i.e.,  $V_i > 8560$ ). This provides evidence for the hypothesis that visibility index calculations can identify a relatively small group of points out of the total set of potential observer locations for further investigation. N37E127-W produces an even smaller number (582) of points with visibility values in the top half of the total range (i.e.,  $V_i > 7316$ ). However, more detailed investigation reveals that points selected from weighted processing tend to cluster together more than in unweighted processing when ranked by visibility index, and can actually require consideration of a larger number of points to achieve a desired minimum number of distinct geographic locations as candidate observer areas.

Another observation to be made from Table C.1 is that magnitude of the correlation between elevation and visibility index is never large, and for the cells shown the value of  $r^2$  for either model never exceeds 0.073684 — hardly a vote of confidence for using either of these elevation-based models to find the best observer points. In some cases the correlation (such as it is) is in fact negative. The results for N36E126 may be heavily influenced by the presence of substantial coastal areas (with relatively low elevations) which have extensive visibility out over the ocean. Whether or not visibility over the ocean is of the same import as visibility over land is a function of the specific application; here they are treated equally.

Achieving marginal results in this initial attempt to use elevation as a guide to determine points with the best visibility, Table C.2 shows the results of applying the same models to the more general problem of relating elevation and visibility for all (non-sea level) points throughout an area. Also shown are some results of the impact of changing the extent of the total area correlated by presenting values for subareas of N37E127-W. Cell names with lowercase letters and suffixed with a lowercase 'w', 'x', 'y', or 'z' are *quadrants* of a complete DTED cell. A 'w' suffix indicates the southwest quadrant, 'x' the northwest quadrant, 'y' the southeast quadrant, and 'z' the northeast quadrant.



Correlation of Visibility to Elevation: All Points			
Cell Name	r Linear Model	r Quadratic Model	Remarks
N37E127	-0.119575	-0.016950	
<b>N37E127-W</b>	<b>0.196629</b>	<b>0.243252</b>	Weighted
N36E127	-0.236553	-0.098787	
N36E128	0.041409	0.11115	
n37e127w-W	0.291576	0.378529	SW Quadrant
n37e127x-W	0.214607	0.263673	NW Quadrant
n37e127y-W	0.257636	0.281329	SE Quadrant
<b>n37e127z-W</b>	<b>0.500397</b>	<b>0.557214</b>	NE Quadrant
<b>n37e127z-W</b>	<b>0.205598</b>	<b>0.170632</b>	164 Points in Top 50% of Vis Range

**Table C.2: All Points Visibility—Elevation Correlations.**

For several complete cells the results are no better — or even significantly worse — than correlations for best visibility points from Table C.1. In some cases (N37E127, N36E127) the sign of the correlation has switched from positive to negative. However, the correlation for N37E127-W for both models has manifested a marked improvement. Selecting this cell for further investigation, subsequent entries in the table show the impact of narrowing the scope of the areas examined to cell quadrants. There is significant variation in the degree of correlation between elevation and visibility among the four quadrants, with n37e127z-W manifesting the highest correlations. For this area, in both the linear and quadratic models nearly 25% of the variation in visibility can be matched to elevation.

It could be hypothesized that by starting with the cell and LOS estimation method yielding the highest initial correlations (N37E127W), examining subareas within it (its quadrants), and selecting the subarea with the highest internal correlation between elevation and visibility (n37e127z-W) that for this restricted set of data it might be possible to obtain useful results in using elevation values to predict the best visibility points. However, as shown in the last entry for Table C.1, the

correlated variation for the 164 points occupying the top 50% of the range of visibility values is still quite low ( $r^2$  of approximately 0.04 or less) and are in fact below the results of Table C.1 using unweighted visibility index values for all of N37E127.

In summary, over the visibility measures, models, and sample areas and sizes presented here, there is no evidence of any straightforward relationships between elevation and visibility that would be useful in quickly selecting most of the best observer sites in a given region. Other models relating elevation to visibility are of course possible, such as those making use of several adjacent elevation values. Given the presence of visibility information for a sufficiently encompassing region, it may be possible to produce a variety of methods more effective than the two models used here through genetic algorithm development or by employing various heuristics. More sophisticated models will of course involve higher computation costs, and in any case would still require a base of visibility estimates or rankings (such as those used here) to evaluate their success in selecting promising observer points.

## APPENDIX D

### Demonstration to User Community

Although the generation of a visibility grid corresponding to the elevation data for an AOI provides the basis for extracting concise information for a variety of siting algorithms, it also offers the opportunity to provide a display to augment existing ways of visually representing terrain considerations for planners and decision makers. Many users of digital terrain data who could provide useful insights into various methods of rendering a visibility image may not have access to workstation environments common in research and academic settings.

In an attempt to make some form of visibility display available to users with relatively restricted computing resources, a standalone demonstration system was developed whose primary screen display is shown in Figure D.1. It can execute on any Intel 80x86 architecture running an MS-DOS or compatible operating system with a standard VGA ( $640 \times 480 \times 16$  color) display. The entire demonstration package is distributed on a single 1.2MB 5.25 inch floppy disk, which can hold the demonstration program, background and operating information text, and elevation and visibility data for 12 areas corresponding to one-degree DTED Level I cells. Also provided for each area are lists of the 16 best and 64 best visible points, as selected from the entire cell and as selected from within 16 equal sub regions with a cell to reduce clustering.

To accommodate the size and color limitations consistent with reaching a large audience, the available elevation (16-bit) and visibility (effectively 12-bit) data are sampled from a  $1201 \times 1201$  grid down to a  $300 \times 300$  grid. Elevation and visibility values are each scaled into thirteen discrete values in a 4-bit nibble, leaving three reserved colors for ocean coloring and two separate highlighting colors for selections of best visibility points. As a result of this reduction, a combined total

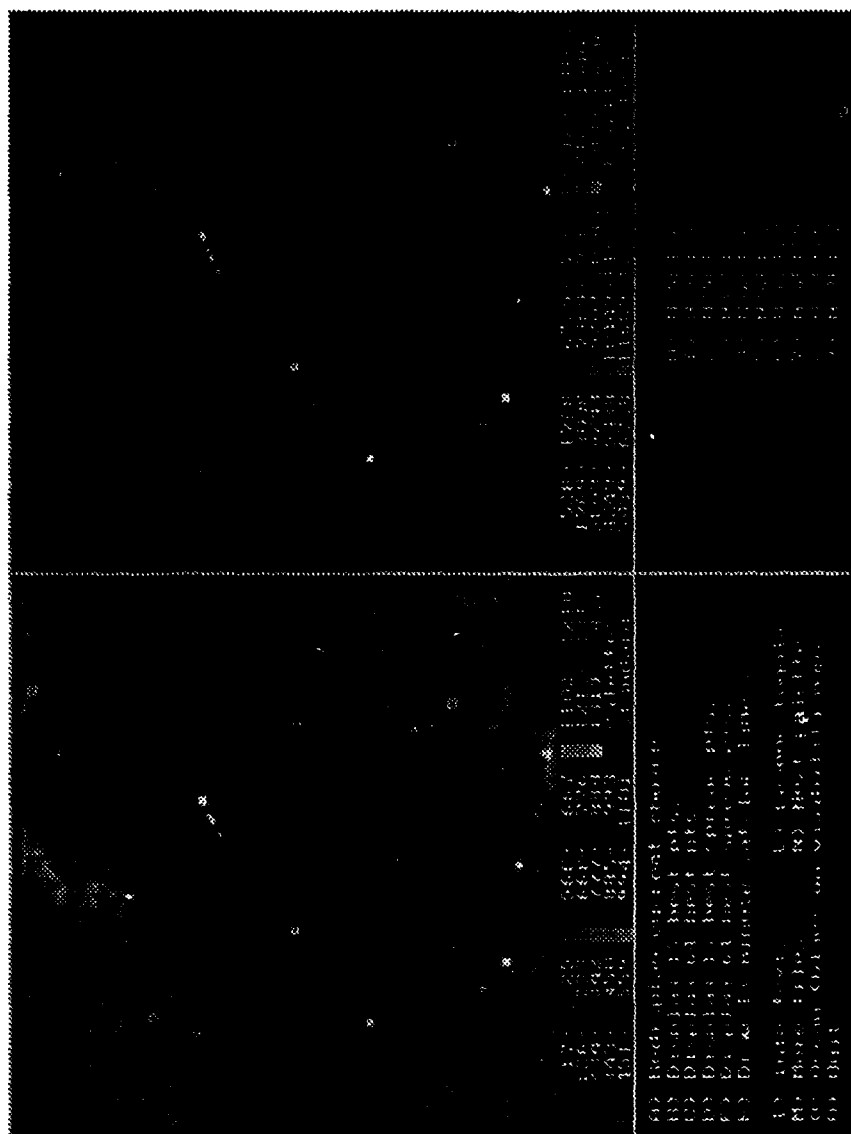


Figure D.1: Screen from PC Demo Application.

of approximately 5.8 MB of elevation and visibility data in one DTED Level I cell is transformed into approximately 90 KB of data for use in the demonstration program. While certainly lacking the detail present in the original data, initial response from the ALBE group at TEC indicates that the reduced data remains useful and representative.

Comments on the initial implementation of the demonstration set led to an enhanced version which provides for legend displays. User configurable palettes were suggested and implemented to allow unanticipated uses of the data, such as highlighting areas of very limited visibility (for concealment) by reversing the normal palette. An application based on this demonstration set is being implemented at TEC in a UNIX X-Window environment to be integrated into the existing ALBE software suite for fielding in the first part of 1994.